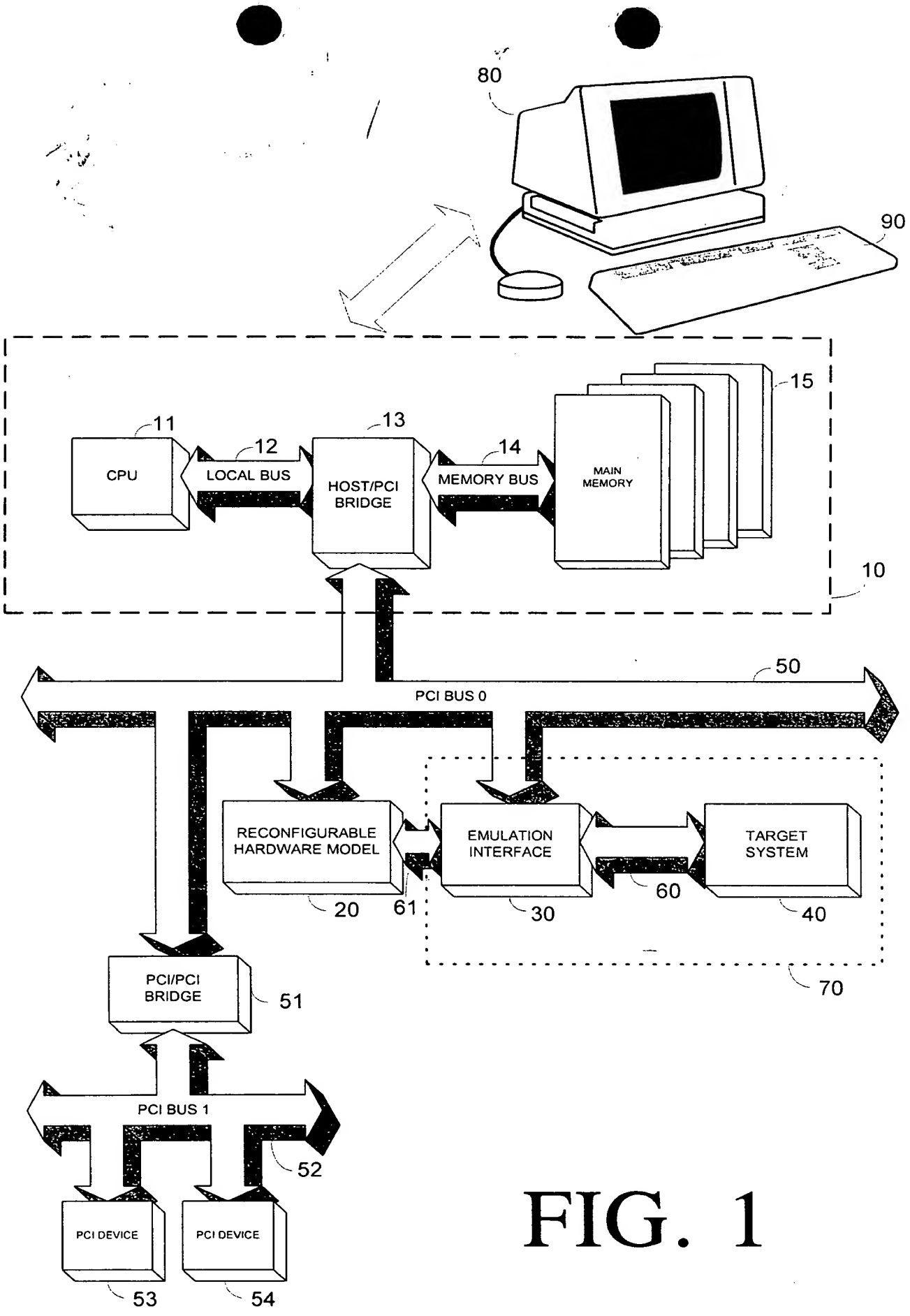

80
90
15
11
13
12
14
CPU
LOCAL BUS
HOST/PCI BRIDGE
MEMORY BUS
MAIN MEMORY
10
50
PCI BUS 0
RECONFIGURABLE HARDWARE MODEL
EMULATION INTERFACE
TARGET SYSTEM
20
61
30
60
40
70
PCI/PCI BRIDGE
51
PCI BUS 1
52
PCI DEVICE
PCI DEVICE
53
54

FIG. 1

USAGE

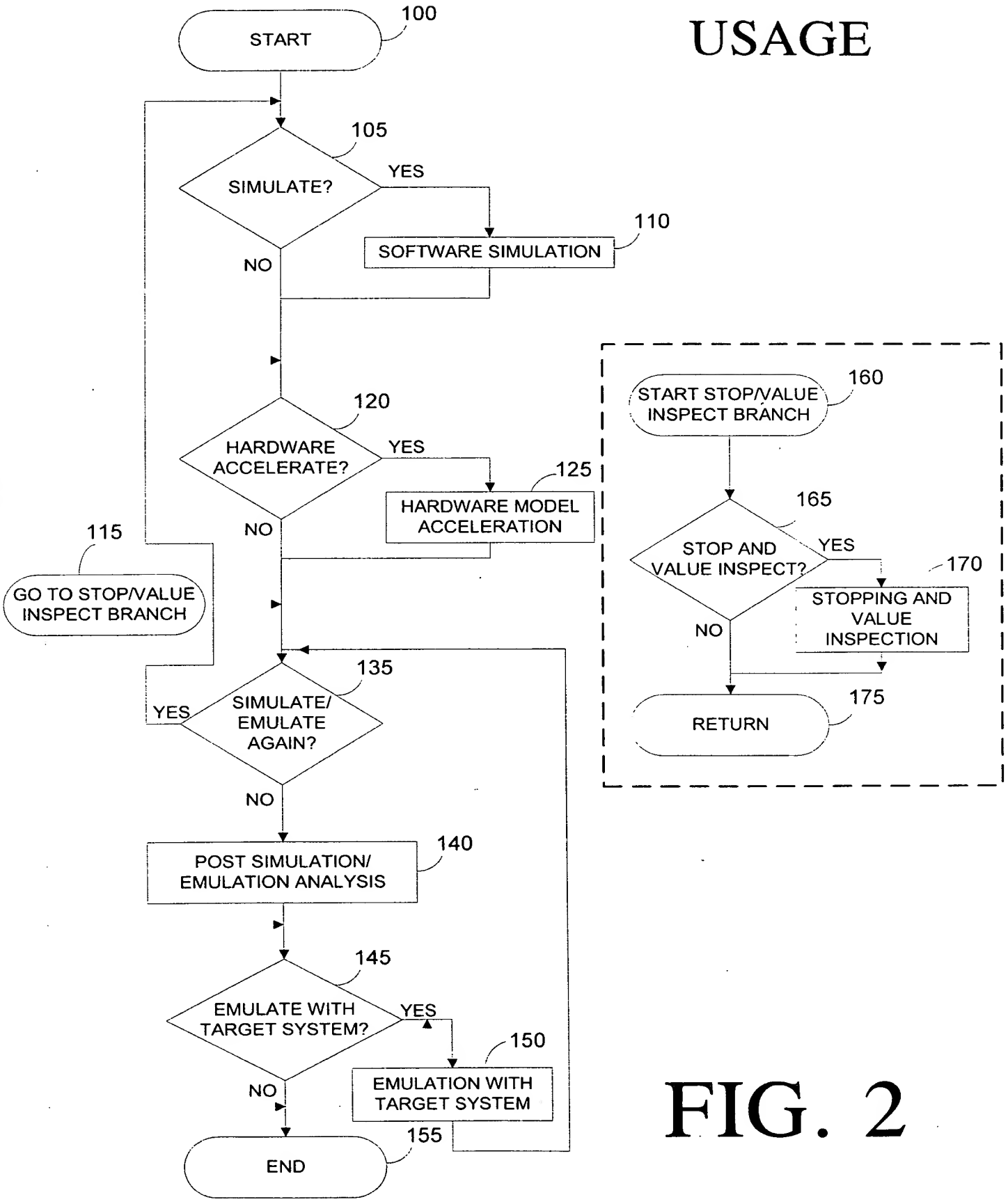

START
100
SIMULATE?
105
YES
NO
SOFTWARE SIMULATION
110
HARDWARE ACCELERATE?
120
YES
NO
HARDWARE MODEL ACCELERATION
125
GO TO STOP/VALUE INSPECT BRANCH
115
SIMULATE/ EMULATE AGAIN?
135
YES
NO
POST SIMULATION/ EMULATION ANALYSIS
140
EMULATE WITH TARGET SYSTEM?
145
YES
NO
EMULATION WITH TARGET SYSTEM
150
END
155
START STOP/VALUE INSPECT BRANCH
160
STOP AND VALUE INSPECT?
165
YES
NO
STOPPING AND VALUE INSPECTION
170
RETURN
175

FIG. 2

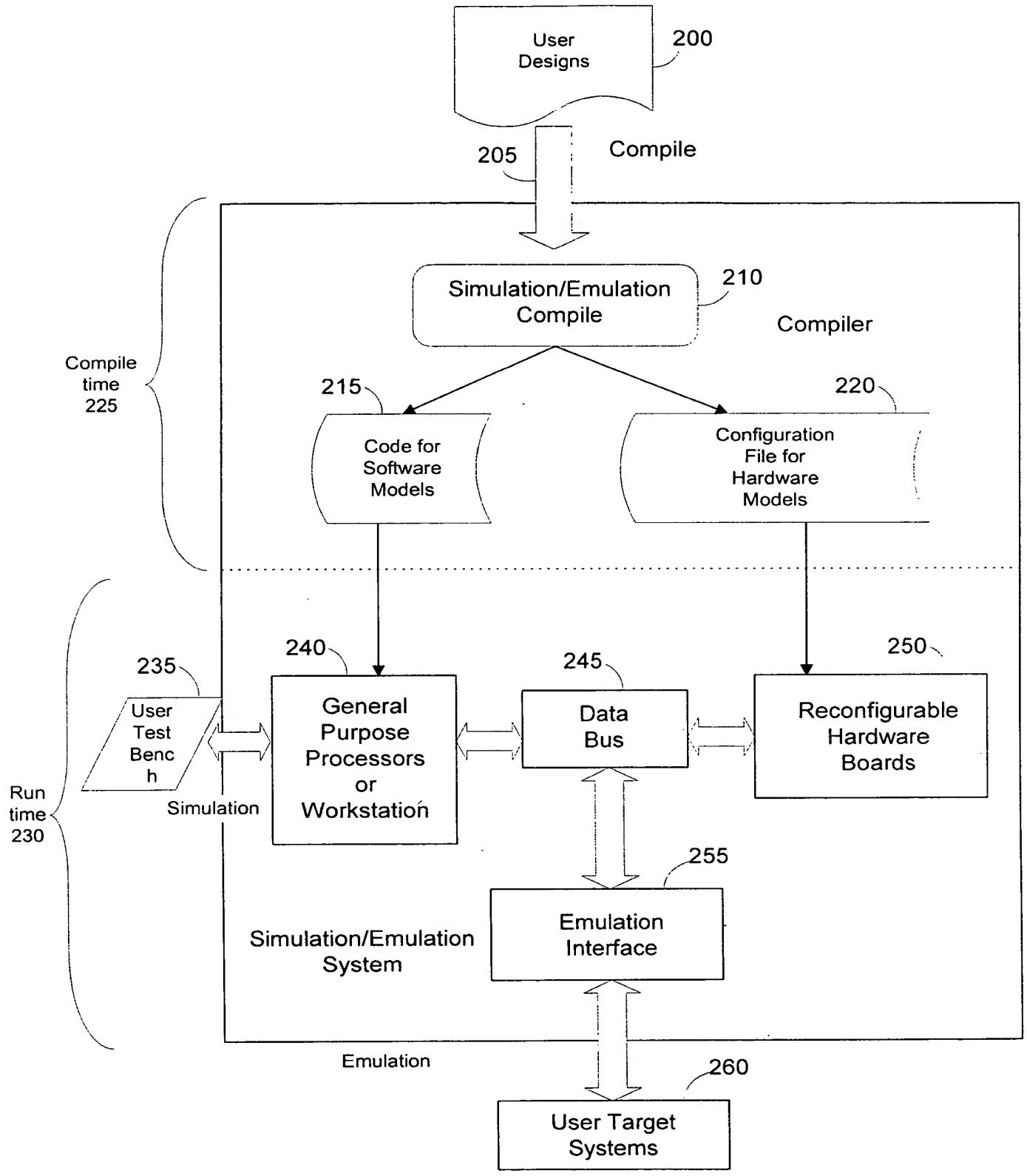

User Designs
200
205
Compile
Simulation/Emulation Compile
210
Compiler
Compile time 225
215
220
Code for Software Models
Configuration File for Hardware Models
235
240
245
250
User Test Benc h
General Purpose Processors or Workstation
Data Bus
Reconfigurable Hardware Boards
Run time 230
Simulation
255
Emulation Interface
Simulation/Emulation System
Emulation
260
User Target Systems

FIG. 3

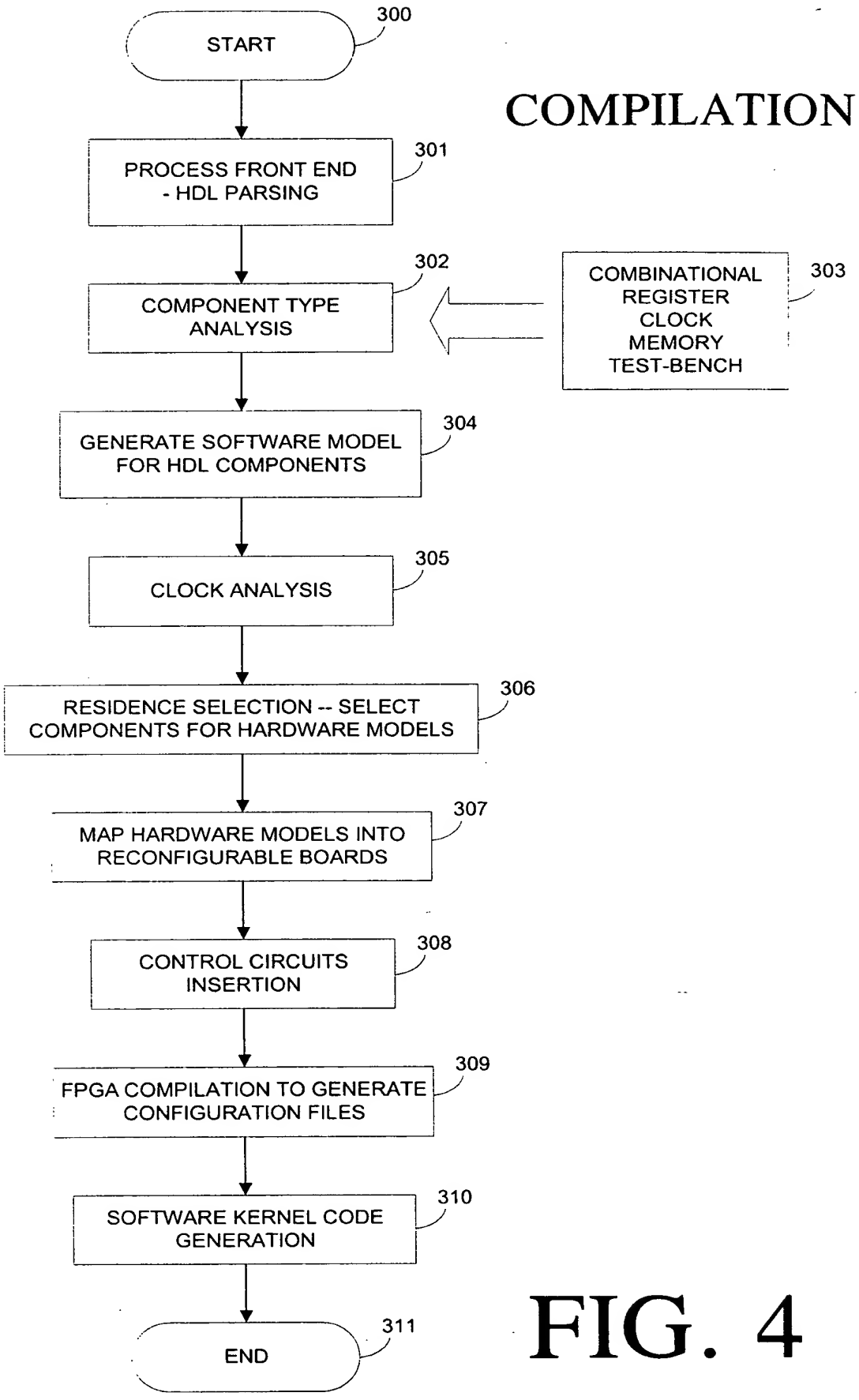
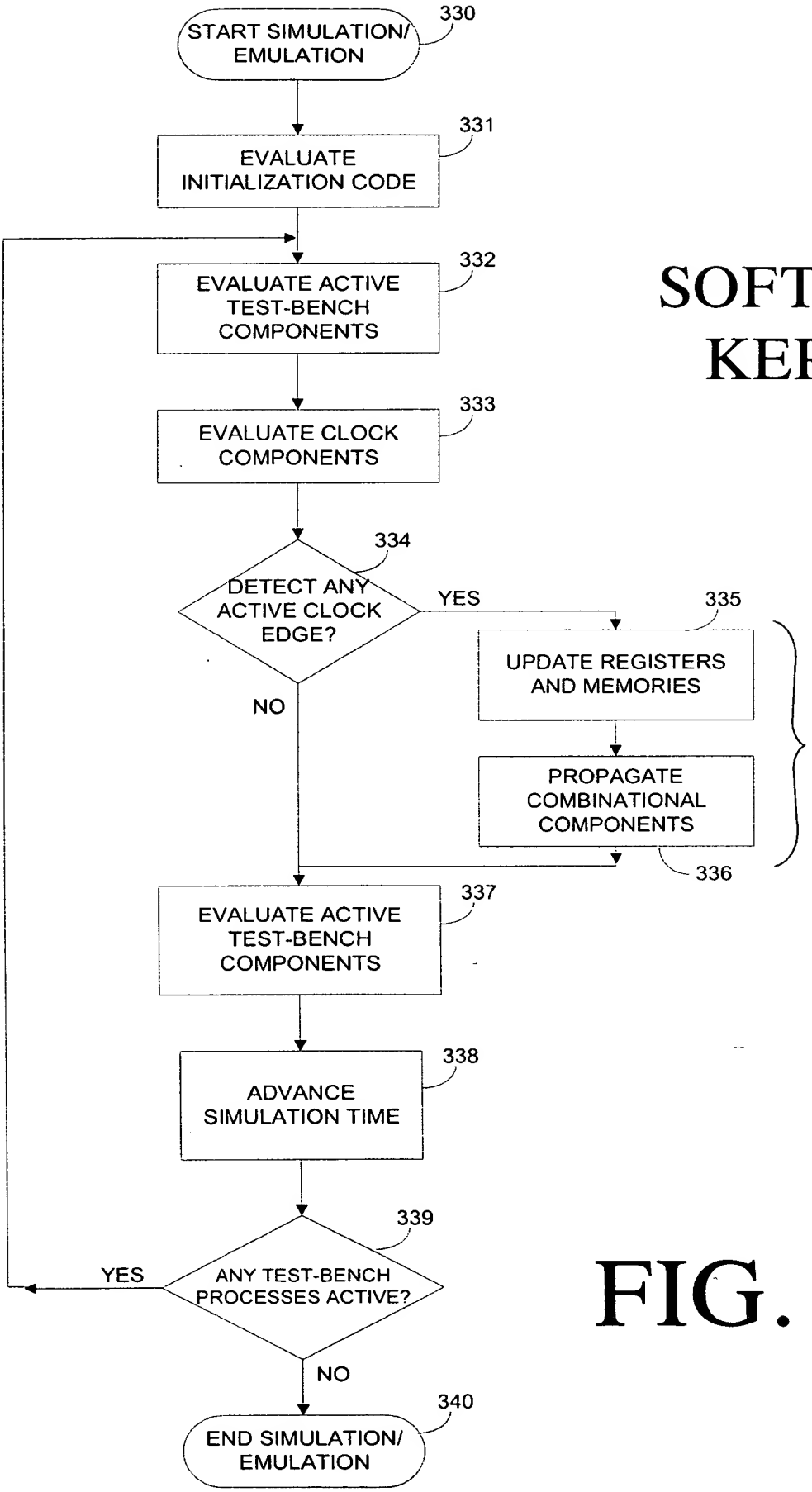

COMPILATION
START
300
PROCESS FRONT END
- HDL PARSING
301
COMPONENT TYPE
ANALYSIS
302
COMBINATIONAL
REGISTER
CLOCK
MEMORY
TEST-BENCH
303
GENERATE SOFTWARE MODEL
FOR HDL COMPONENTS
304
CLOCK ANALYSIS
305
RESIDENCE SELECTION -- SELECT
COMPONENTS FOR HARDWARE MODELS
306
MAP HARDWARE MODELS INTO
RECONFIGURABLE BOARDS
307
CONTROL CIRCUITS
INSERTION
308
FPGA COMPILATION TO GENERATE
CONFIGURATION FILES
309
SOFTWARE KERNEL CODE
GENERATION
310
END
311

FIG. 4



SOFTWARE KERNEL

FIG. 5

MAPPING HARDWARE MODELS TO RECONFIGURABLE BOARDS

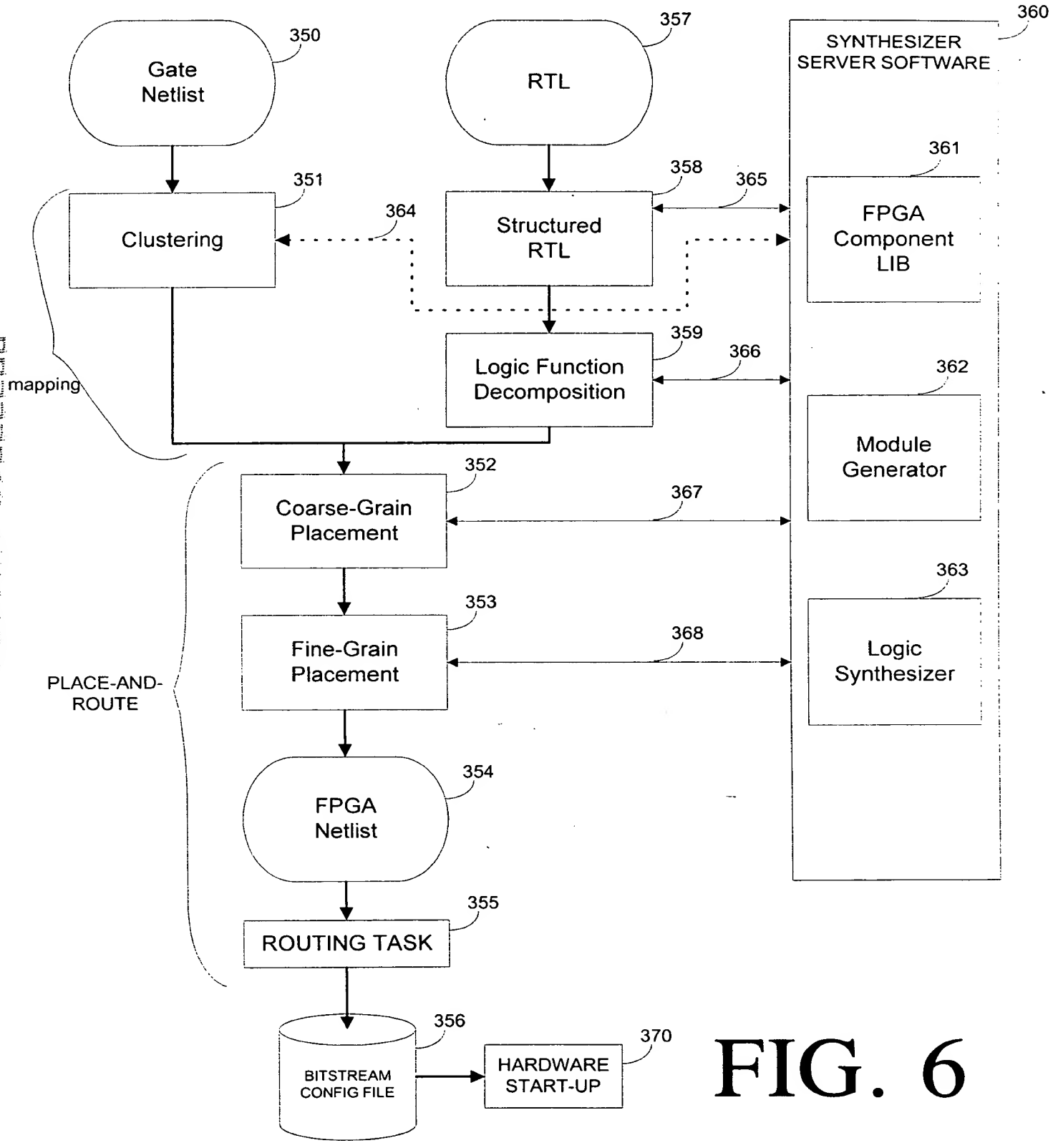


FIG. 6

	F11	F12	F13	F14	F21	F22	F23	F24	F31	F32	F33	F34	F41	F42	F43	F44
F11	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0
F12	1	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0
F13	1	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0
F14	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1
F21	0	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0
F22	1	1	0	0	1	1	1	1	0	1	0	0	0	1	0	0
F23	0	0	1	0	1	1	1	1	0	0	1	0	0	0	1	0
F24	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1
F31	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0
F32	1	1	0	0	0	1	0	0	1	1	1	1	0	1	0	0
F33	0	0	1	0	0	0	1	0	1	1	1	1	0	0	1	0
F34	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
F41	0	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1
F42	1	1	0	0	0	1	0	0	0	1	0	0	1	1	1	1
F43	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	1
F44	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1

FIG. 7

FPGA INTERCONNECTION

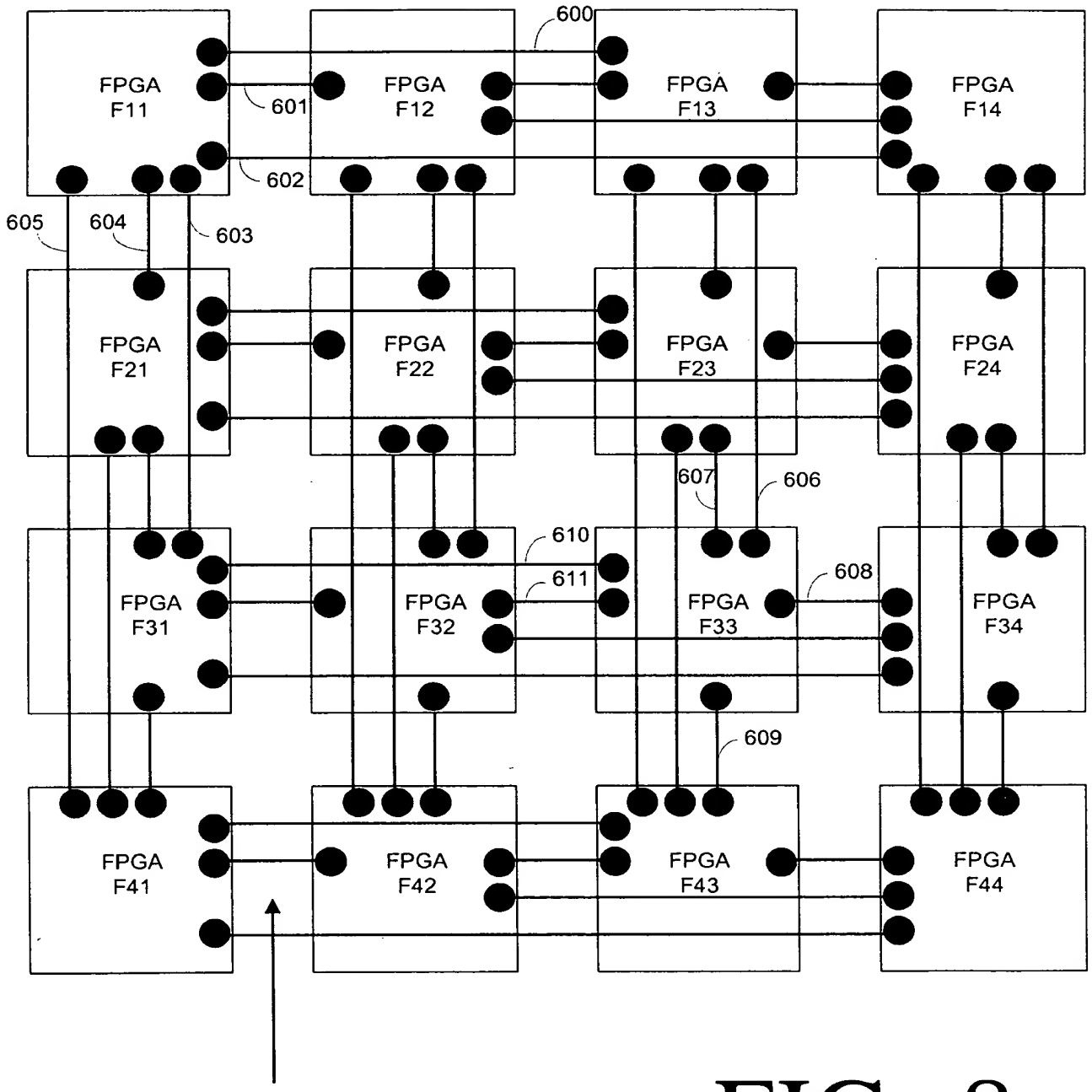


FIG. 8

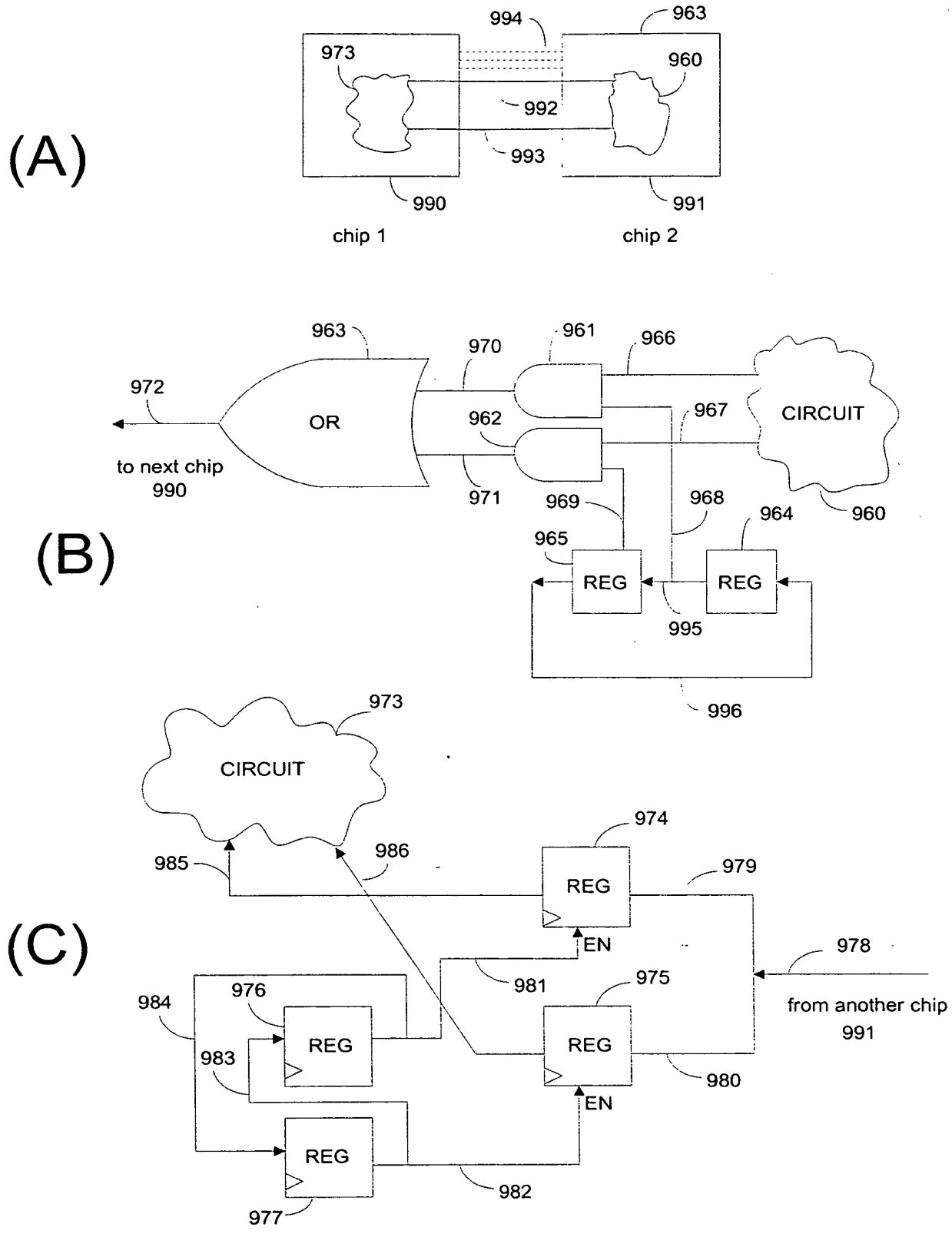

(A)
994
963
973
960
992
993
990
991
chip 1
chip 2
(B)
963
970
961
966
972
OR
962
967
CIRCUIT
to next chip
990
971
969
968
964
960
965
REG
REG
995
996
(C)
973
CIRCUIT
974
985
986
REG
979
EN
978
981
975
976
984
REG
REG
from another chip
991
983
980
EN
REG
982
977

FIG. 9

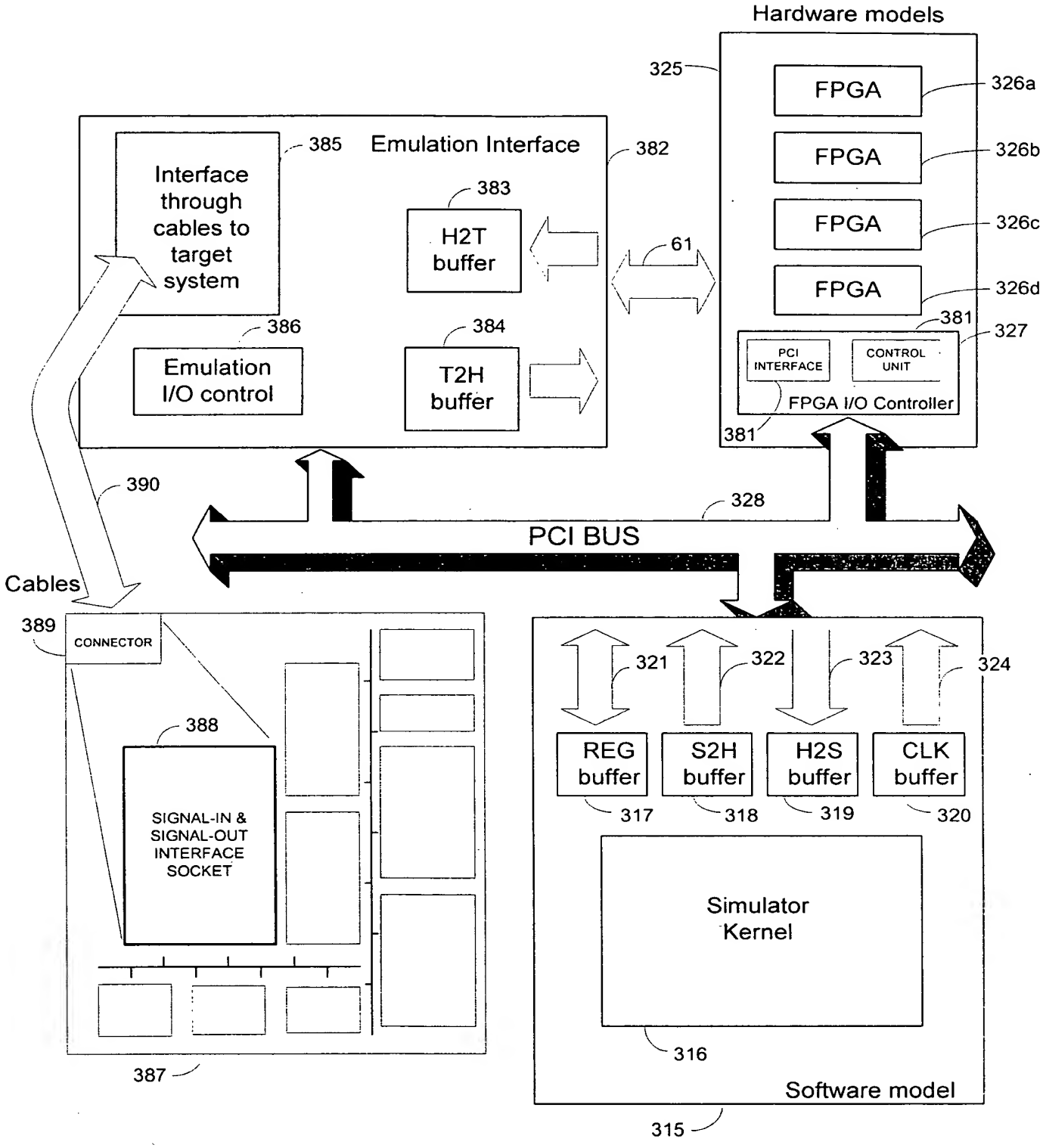

Hardware models
325
FPGA
326a
FPGA
326b
FPGA
326c
FPGA
326d
381
327
PCI INTERFACE
CONTROL UNIT
FPGA I/O Controller
381
Interface through cables to target system
385
Emulation Interface
382
383
H2T buffer
61
386
Emulation I/O control
384
T2H buffer
390
328
PCI BUS
Cables
389
CONNECTOR
388
SIGNAL-IN & SIGNAL-OUT INTERFACE SOCKET
387
321
322
323
324
REG buffer
S2H buffer
H2S buffer
CLK buffer
317
318
319
320
Simulator Kernel
316
Software model
315

FIG. 10

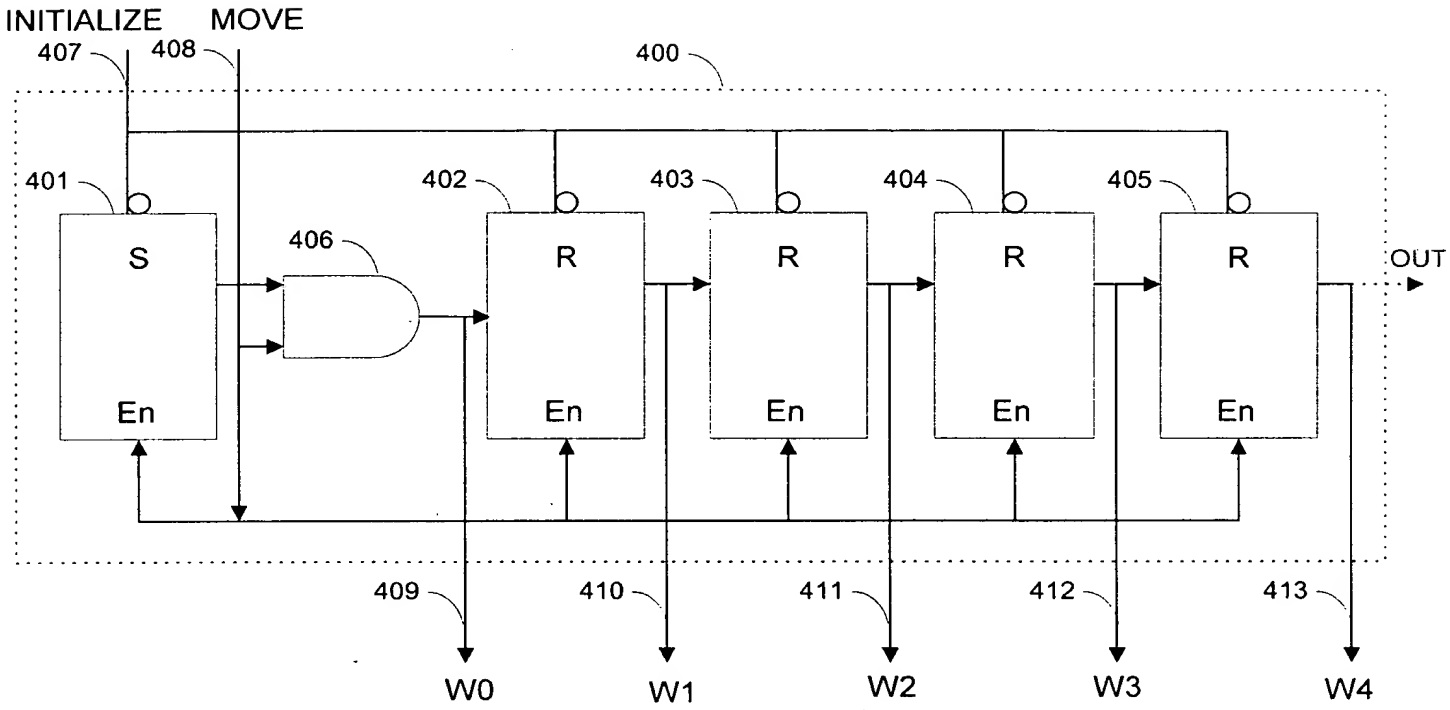

INITIALIZE
MOVE
407
408
400
401
402
403
404
405
406
S
R
R
R
R
En
En
En
En
En
OUT
409
410
411
412
413
W0
W1
W2
W3
W4

FIG. 11

ADDRESS POINTER INITIALIZATION

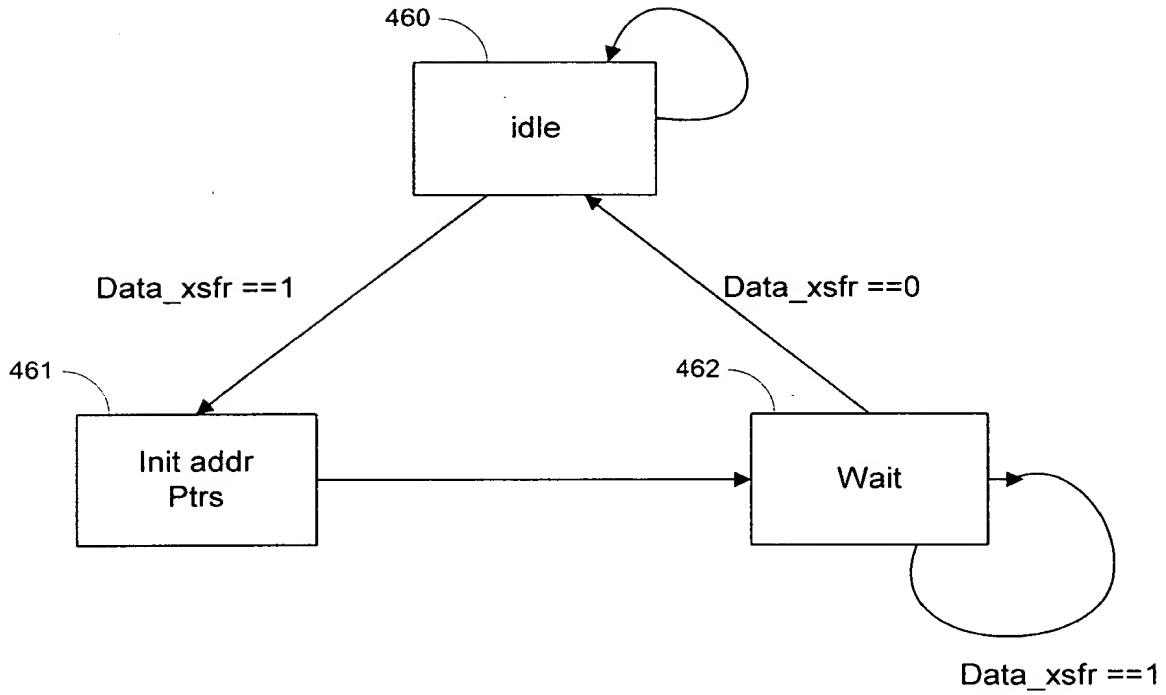


FIG. 12

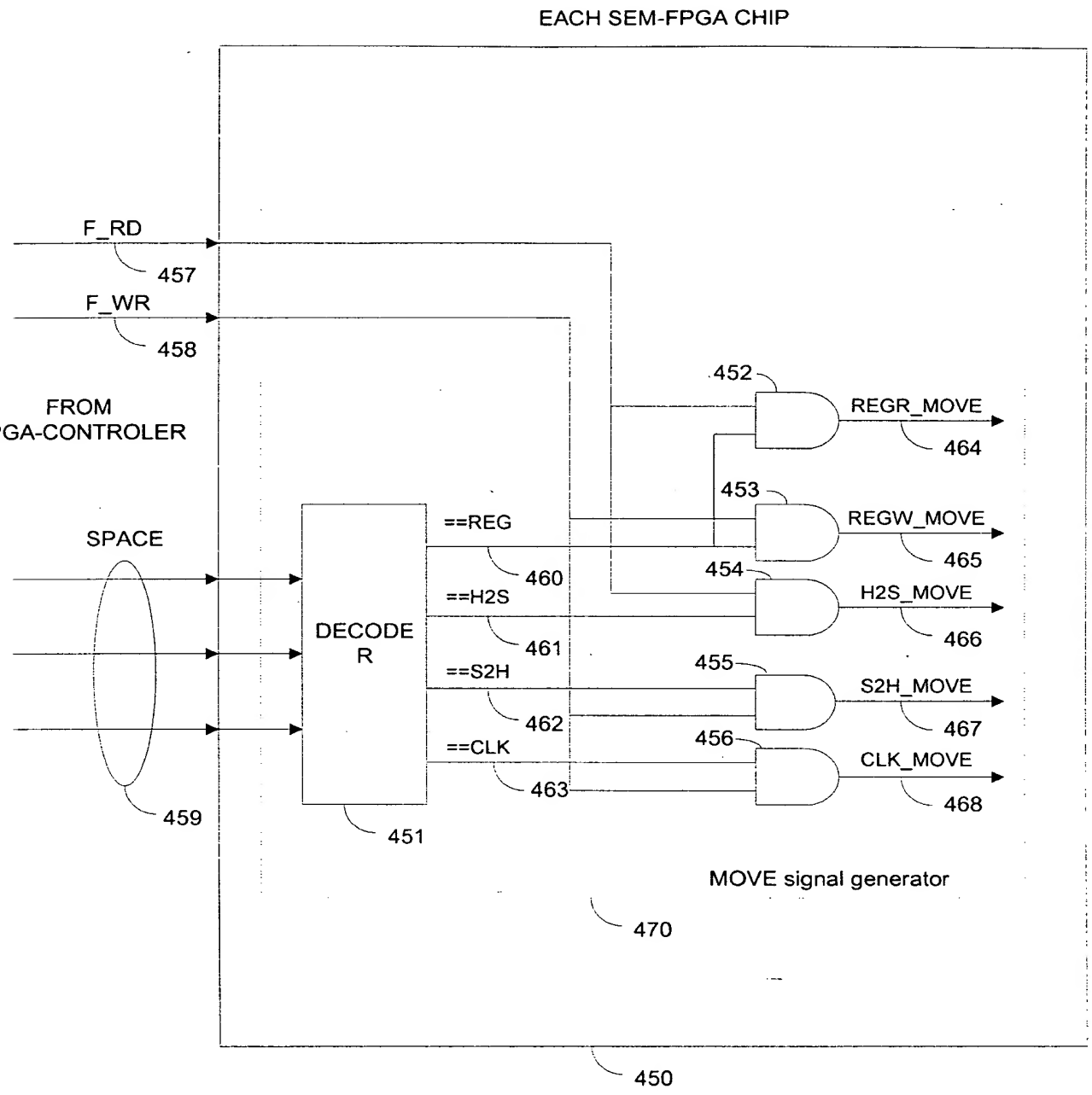

EACH SEM-FPGA CHIP
F_RD
457
F_WR
458
FROM
FPGA-CONTROLER
SPACE
459
DECODE
R
451
==REG
460
==H2S
461
==S2H
462
==CLK
463
452
REGR_MOVE
464
453
REGW_MOVE
465
454
H2S_MOVE
466
455
S2H_MOVE
467
456
CLK_MOVE
468
MOVE signal generator
470
450

FIG. 13

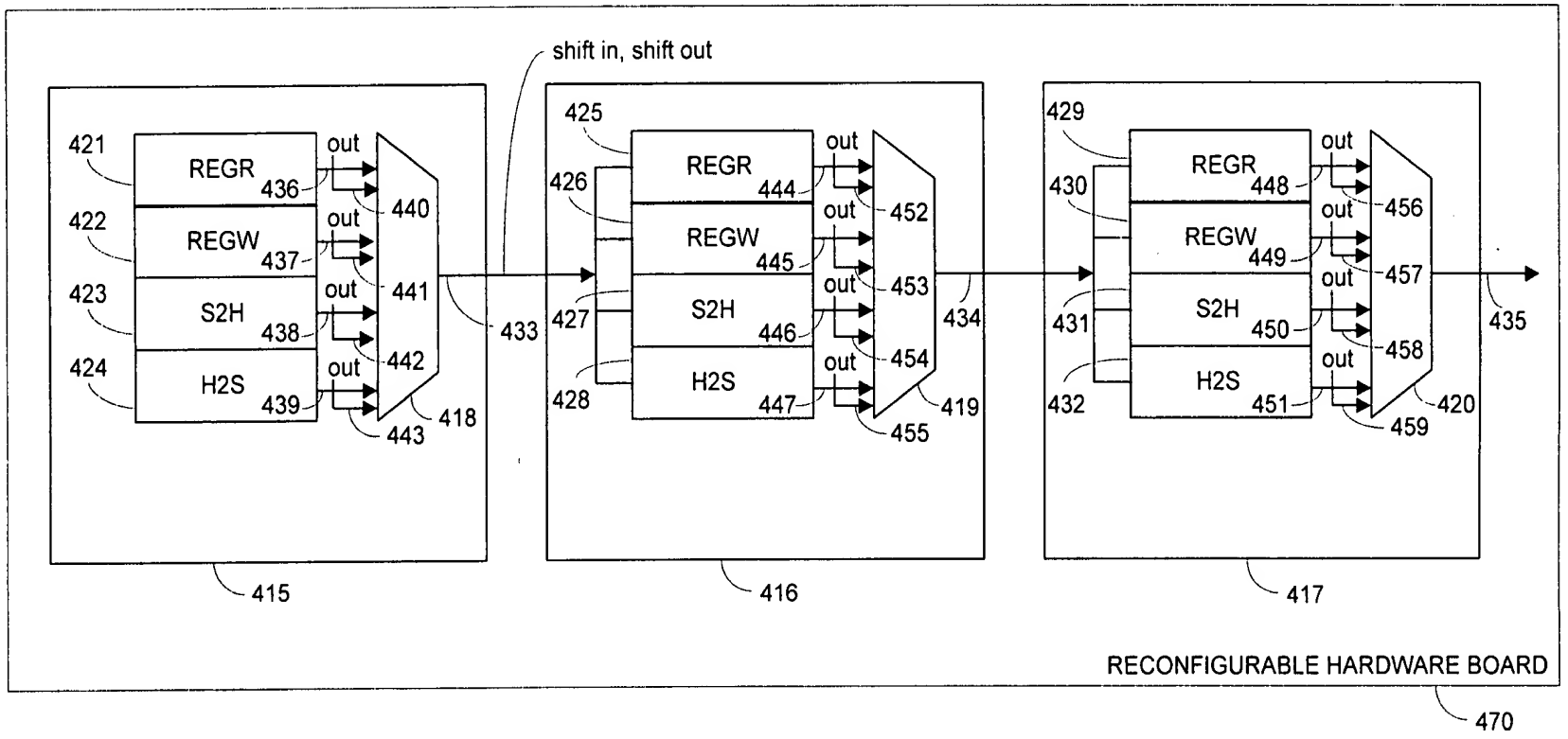

shift in, shift out
421
REGR
436
out
422
REGW
437
out
423
S2H
438
out
424
H2S
439
out
440
441
442
443
418
433
415
425
426
427
428
REGR
444
REGW
445
S2H
446
H2S
447
out
452
453
454
455
419
434
416
429
430
431
432
REGR
448
REGW
449
S2H
450
H2S
451
out
456
457
458
459
420
435
417
RECONFIGURABLE HARDWARE BOARD
470

FIG. 14

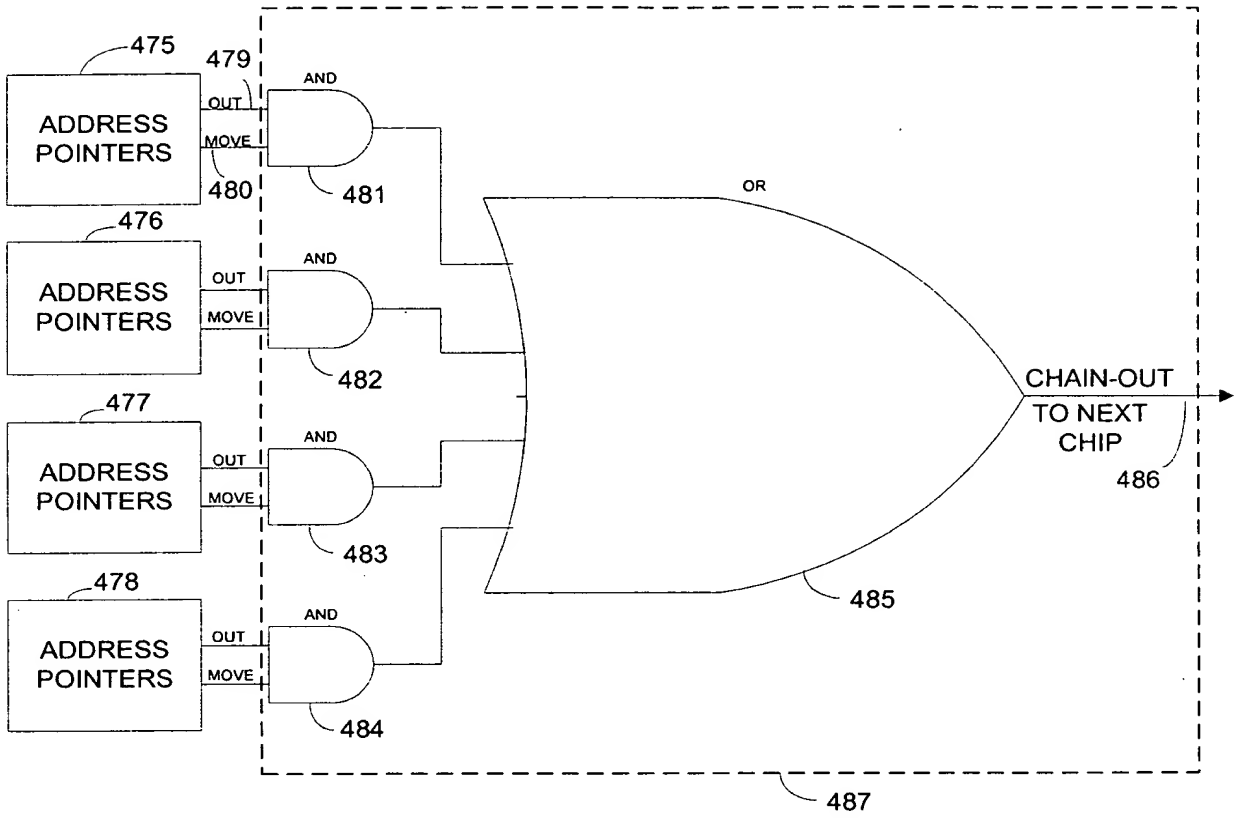

475
479
ADDRESS POINTERS
OUT
MOVE
480
AND
481
476
ADDRESS POINTERS
OUT
MOVE
AND
482
477
ADDRESS POINTERS
OUT
MOVE
AND
483
478
ADDRESS POINTERS
OUT
MOVE
AND
484
OR
485
CHAIN-OUT
TO NEXT CHIP
486
487

FIG. 15

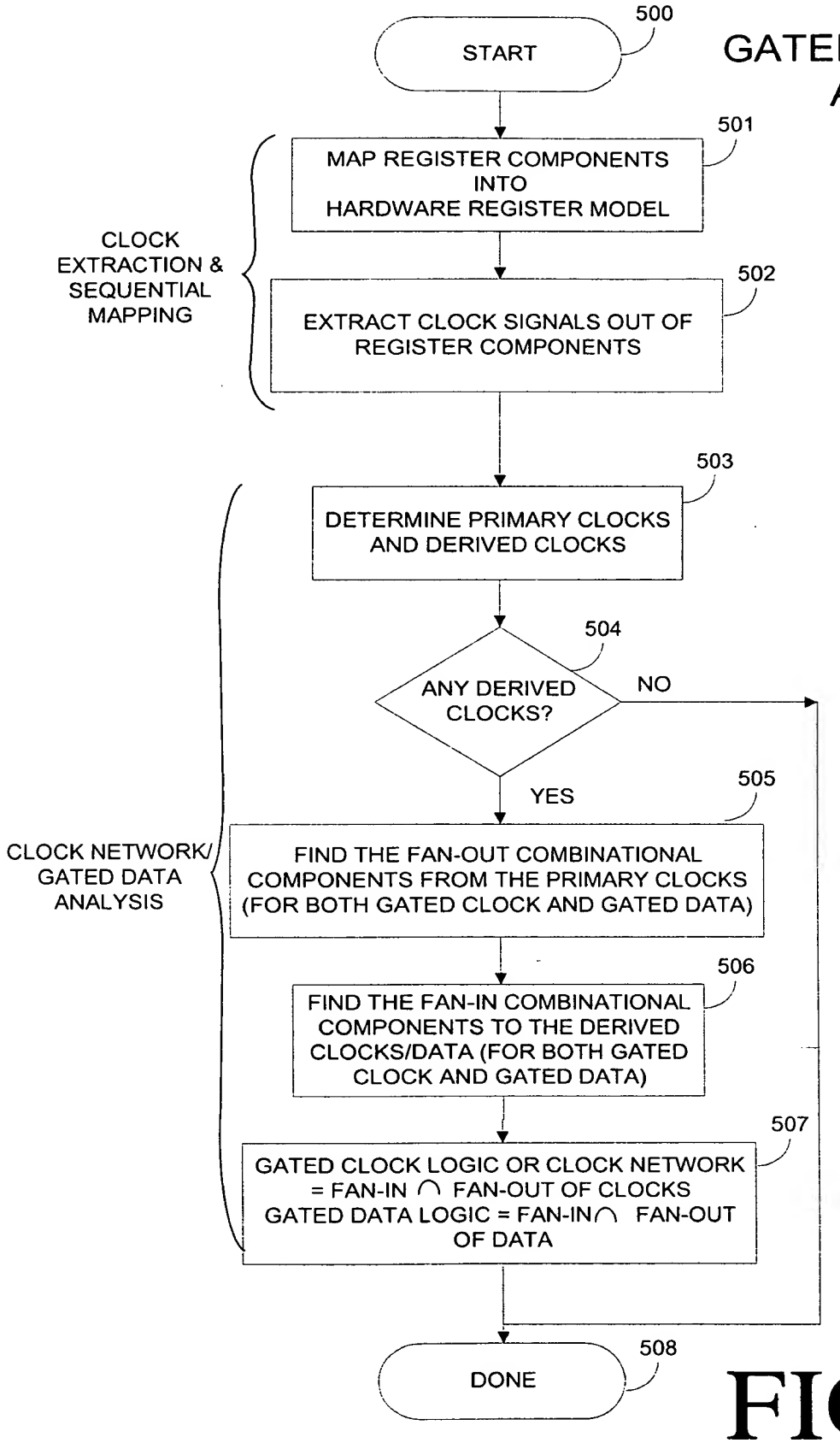


FIG. 16

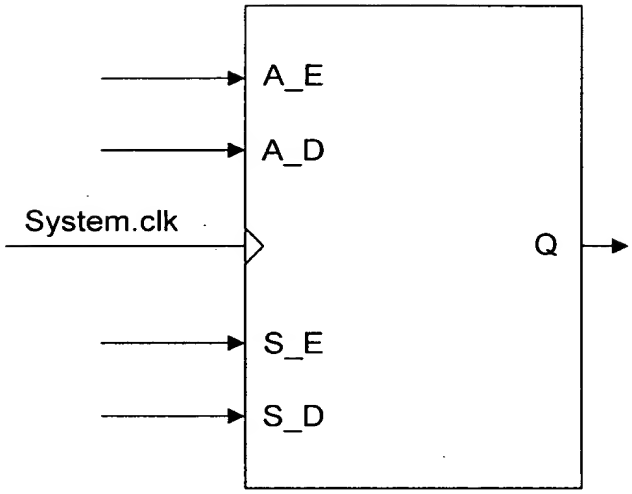
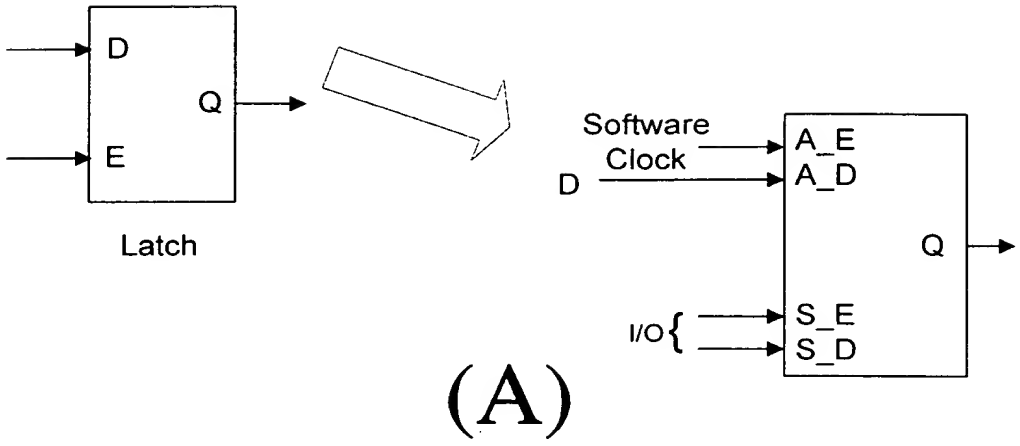
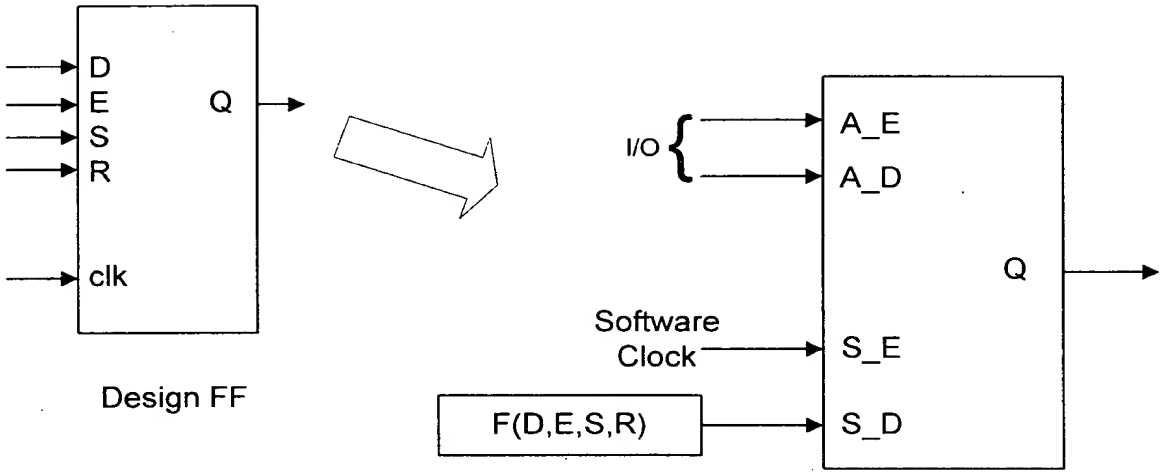


FIG. 17


D
Q
E
Latch
Software
Clock
D
A_E
A_D
Q
I/O
S_E
S_D

(A)


D
E
S
R
Q
clk
Design FF
I/O
A_E
A_D
Q
Software
Clock
S_E
F(D,E,S,R)
S_D

(B)

FIG. 18

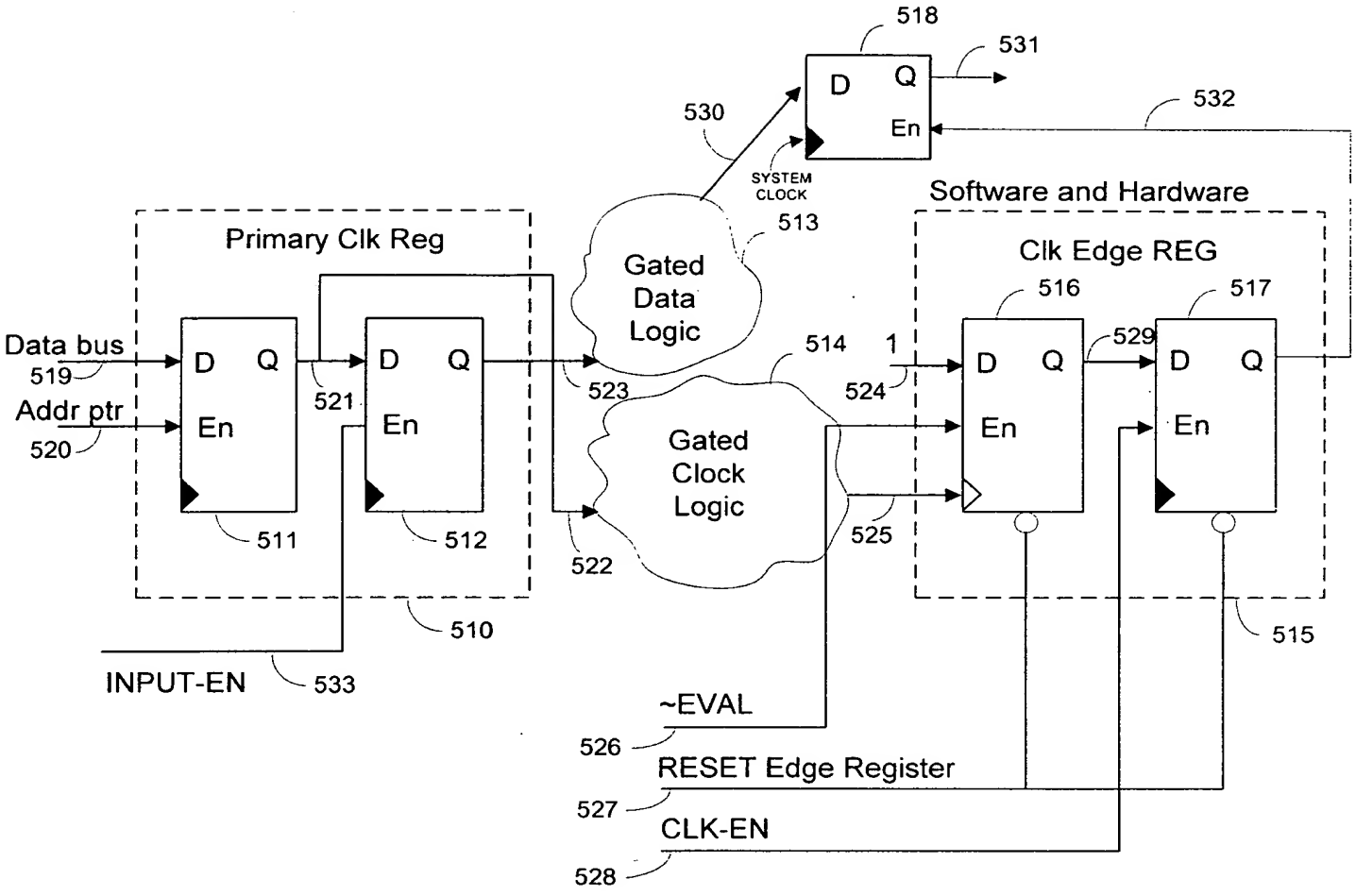

518
531
D
Q
En
530
532
SYSTEM CLOCK
Software and Hardware
513
Primary Clk Reg
Gated Data Logic
Clk Edge REG
516
517
Data bus
519
Addr ptr
520
D
Q
En
521
514
1
524
529
523
Gated Clock Logic
522
525
511
512
510
515
INPUT-EN
533
~EVAL
526
RESET Edge Register
527
CLK-EN
528

FIG. 19

DURING
EVALUATION

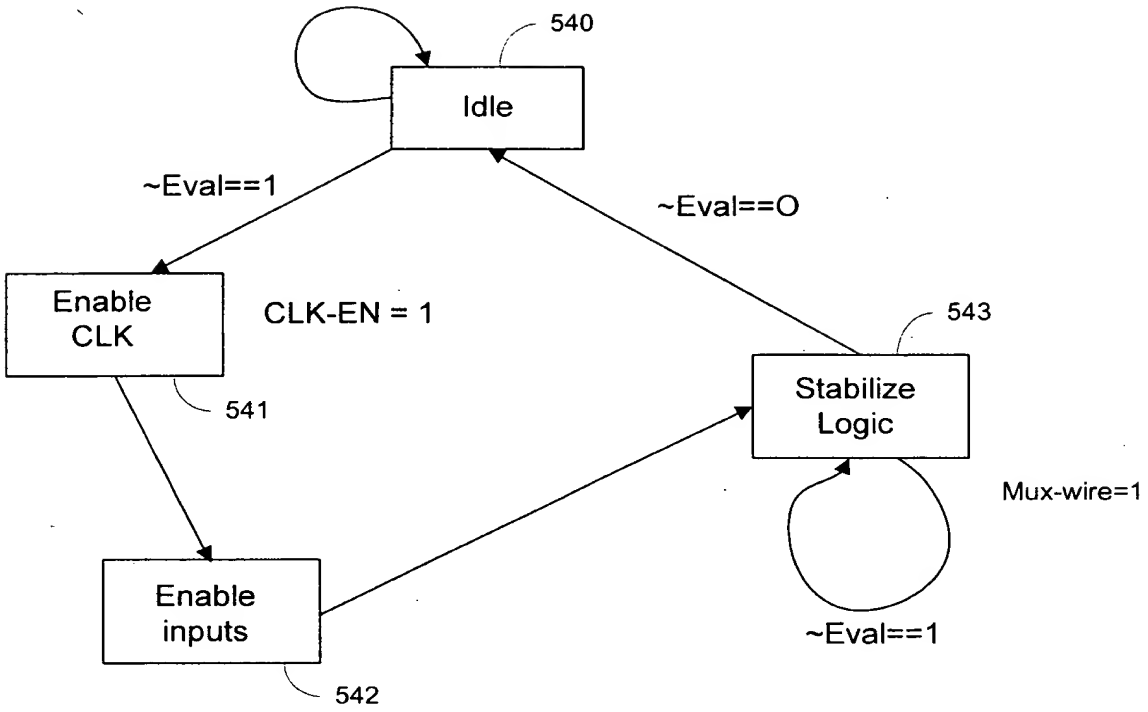


FIG. 20

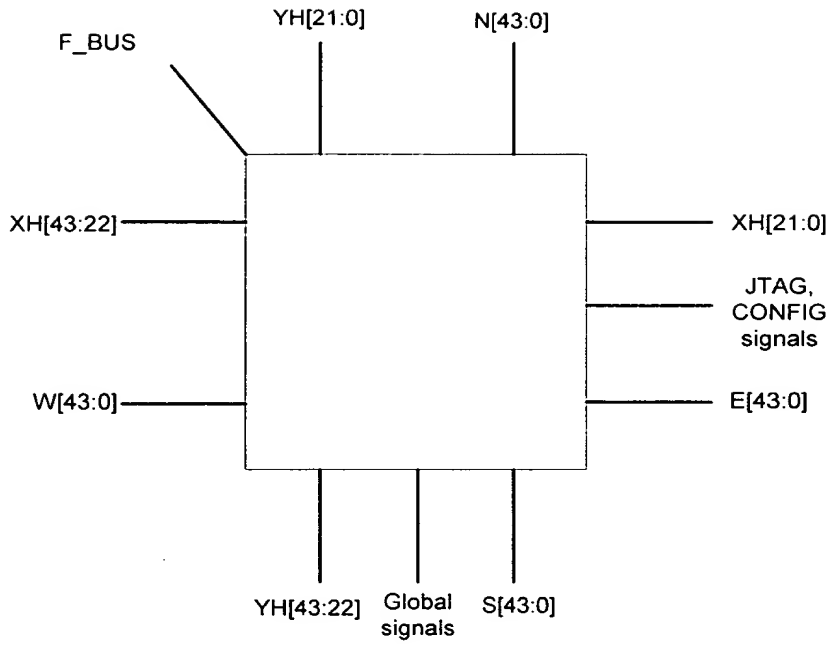

F_BUS
YH[21:0]
N[43:0]
XH[43:22]
XH[21:0]
JTAG, CONFIG signals
W[43:0]
E[43:0]
YH[43:22]
Global signals
S[43:0]

FIG. 21

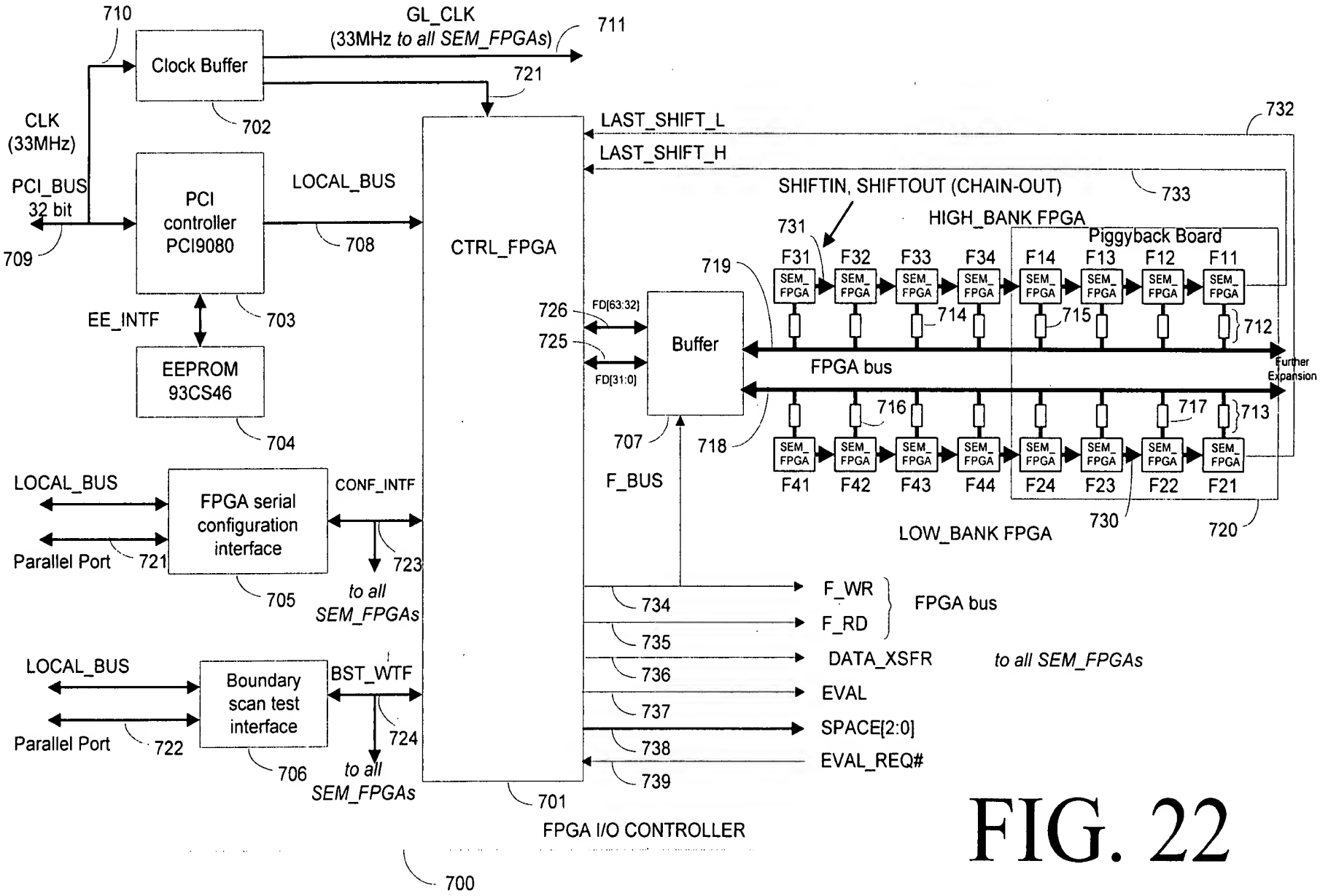

710
Clock Buffer
GL_CLK
(33MHz to all SEM_FPGAs)
711
721
702
CLK
(33MHz)
PCI_BUS
32 bit
709
PCI
controller
PCI9080
LOCAL_BUS
708
CTRL_FPGA
EE_INTF
703
EEPROM
93CS46
704
LAST_SHIFT_L
LAST_SHIFT_H
732
SHIFTIN, SHIFTOUT (CHAIN-OUT)
733
731
HIGH_BANK FPGA
Piggyback Board
719
F31
F32
F33
F34
F14
F13
F12
F11
SEM_FPGA
714
715
712
726
FD[63:32]
725
FD[31:0]
Buffer
FPGA bus
Further Expansion
716
717
713
707
718
F_BUS
F41
F42
F43
F44
F24
F23
F22
F21
730
720
LOW_BANK FPGA
LOCAL_BUS
Parallel Port
721
FPGA serial
configuration
interface
CONF_INTF
723
705
to all
SEM_FPGAs
734
F_WR
F_RD
FPGA bus
735
DATA_XSFR
to all SEM_FPGAs
736
EVAL
737
SPACE[2:0]
738
EVAL_REQ#
739
LOCAL_BUS
Parallel Port
722
Boundary
scan test
interface
BST_WTF
724
706
to all
SEM_FPGAs
701
FPGA I/O CONTROLLER
700

FIG. 22

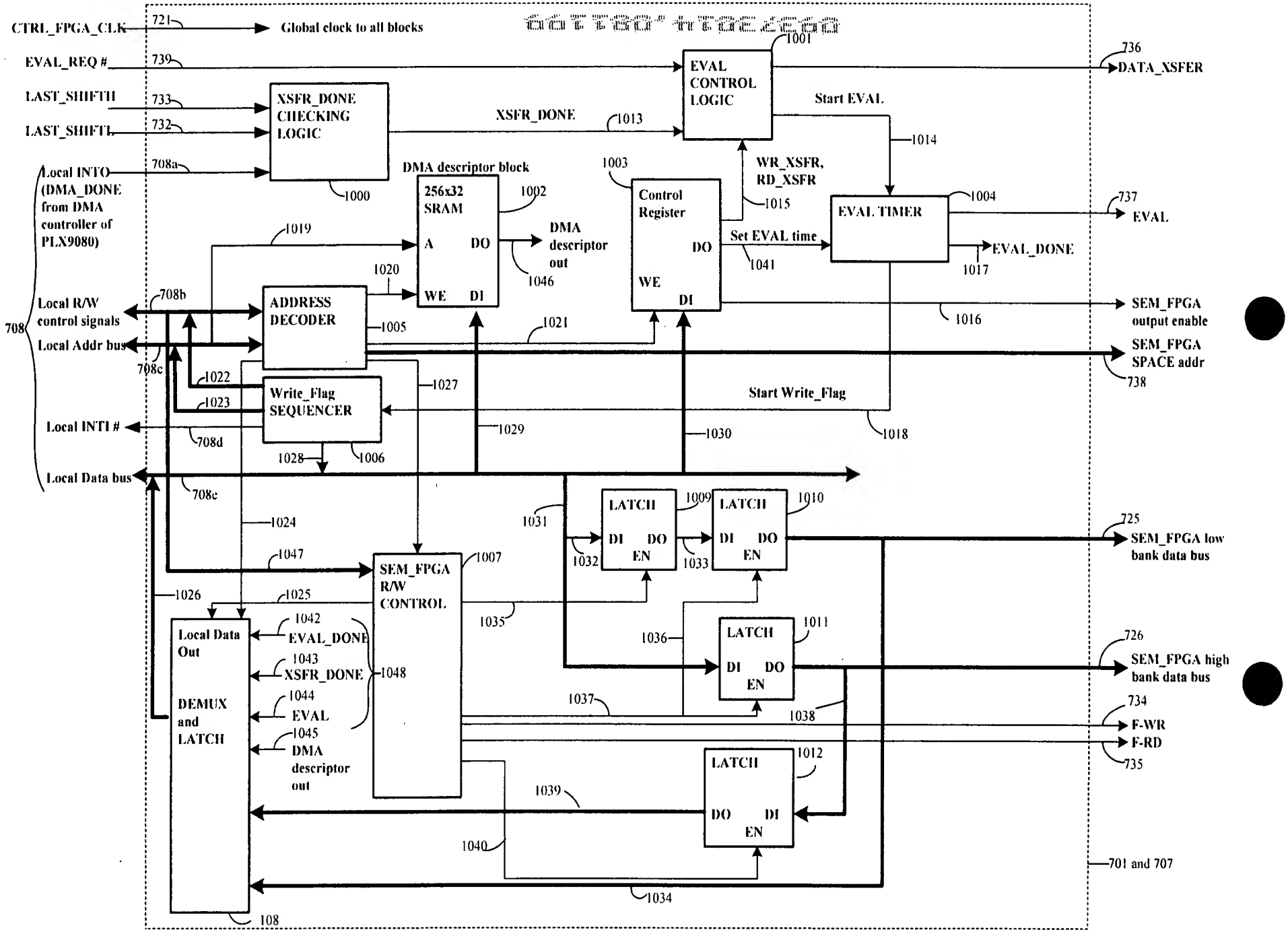

CTRL_FPGA_CLK
721
Global clock to all blocks
EVAL_REQ #
739
LAST_SHIFTH
733
LAST_SHIFTL
732
XSFR_DONE CHECKING LOGIC
1000
XSFR_DONE
1013
EVAL CONTROL LOGIC
1001
736
DATA_XSFER
Start EVAL
1014
WR_XSFR, RD_XSFR
1015
Local INT0 (DMA_DONE from DMA controller of PLX9080)
708a
DMA descriptor block
256x32 SRAM
A DO
WE DI
1002
DMA descriptor out
1046
1003
Control Register
DO
WE
DI
Set EVAL time
1041
EVAL TIMER
1004
737
EVAL
EVAL_DONE
1017
1019
1020
708
Local R/W control signals
Local Addr bus
708b
708c
ADDRESS DECODER
1005
1021
1016
SEM_FPGA output enable
SEM_FPGA SPACE addr
738
1022
1023
Write_Flag SEQUENCER
1027
Start Write_Flag
1018
Local INT1 #
708d
1029
1030
1028
1006
Local Data bus
708e
1024
1031
LATCH
DI DO
EN
1009
1032
1033
LATCH
1010
725
SEM_FPGA low bank data bus
1047
SEM_FPGA R/W CONTROL
1007
1026
1025
1035
1036
LATCH
1011
726
SEM_FPGA high bank data bus
Local Data Out
DEMUX and LATCH
1042
EVAL_DONE
1043
XSFR_DONE
1044
EVAL
1045
DMA descriptor out
1048
1037
1038
734
F-WR
F-RD
735
LATCH
1012
1039
DO DI
EN
1040
701 and 707
1034
108

FIG. 23

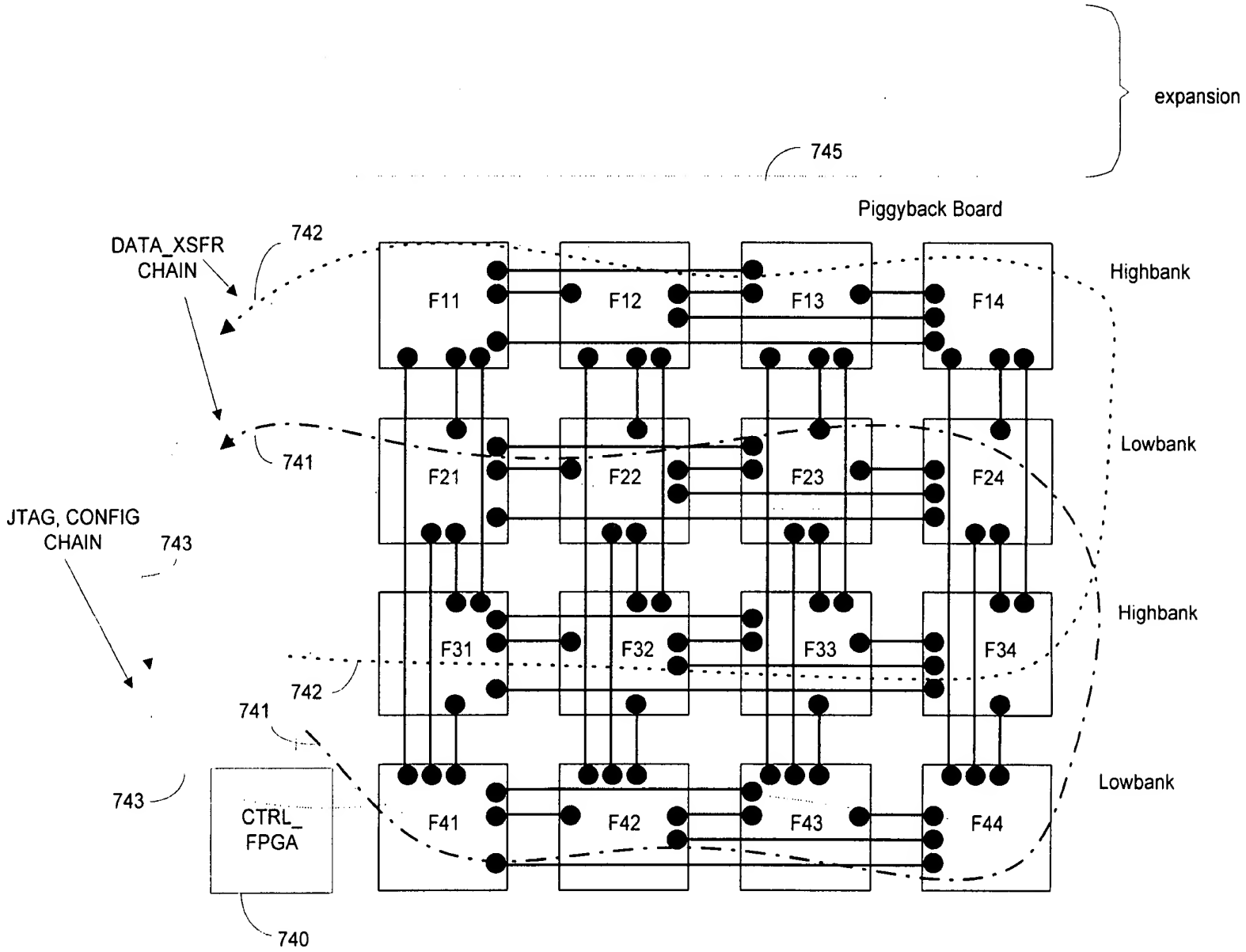

expansion
745
Piggyback Board
DATA_XSFR
CHAIN
742
Highbank
F11
F12
F13
F14
741
Lowbank
F21
F22
F23
F24
JTAG, CONFIG
CHAIN
743
Highbank
F31
F32
F33
F34
742
741
743
Lowbank
CTRL_
FPGA
F41
F42
F43
F44
740

FIG. 24

HARDWARE START-UP

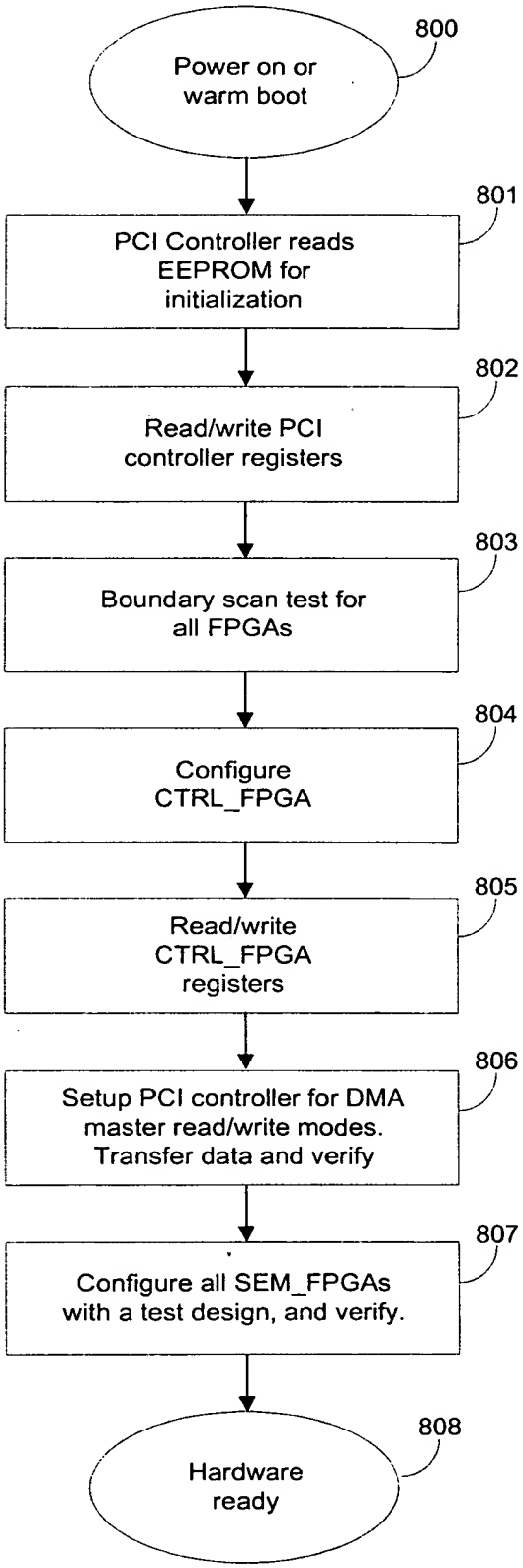


FIG. 25

```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(posedge clock or negedge reset)
    if(!reset)
        q = 0;
    else
        q = d;

endmodule

module example;
    wire d1, d2, d3;
    wire q1, q2, q3;

    reg sigin;
    wire sigout;
    reg clk, reset;

    register reg1 (clk, reset, d1, q1);
    register reg2 (clk, reset, d2, q2);
    register reg3 (clk, reset, d3, q3);

    assign d1 = sigin ^ q3;
    assign d2 = q1 ^ q3;
    assign d3 = q2 ^ q3;
    assign sigout = q3;

    // a clock generator
    always
    begin
        clk = 0;
        #5;
        clk = 1;
        #5;
    end

    // a signal generator
    always
    begin
        #10;
        sigin = $random;
    end

    // initialization
    initial
    begin
        reset = 0;
        sigin = 0;
        #1;
        reset = 1;
        #5;
        $monitor($time, " %b, %b", sigin, sigout);
        #1000 $finish;
    end
    end module
```

FIG. 26

CIRCUIT DIAGRAM

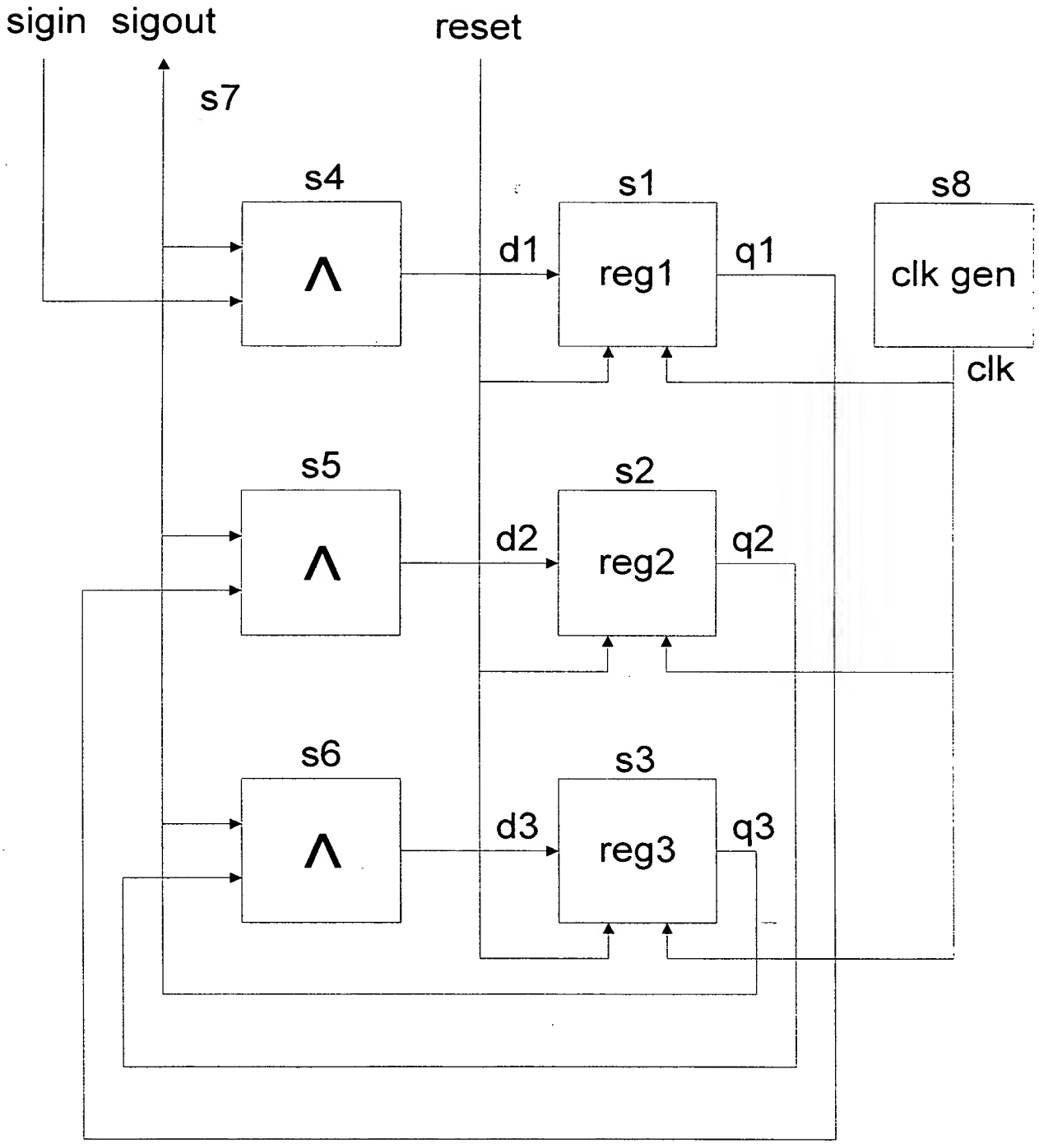


FIG. 27

```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(postedge clock or negedge reset)
   if(~reset)
      q = 0
   else
      q = d;

endmodule
```

Register Definition — 900

```
module example;
   wire d1, d2, d3;
   ware q1, q2, q3;
```

wire interconnection info — 907

```
   reg sigin;
   wire sigout;
   reg clk, reset;
```

Test-bench input -- 908

Test-bench output -- 909

```
S1  register reg 1 (clk, reset, d1, q1);
S2  register reg 2 (clk, reset, d2, q2);
S3  register reg 3 (clk, reset, d3, q3);
```

Register component — 901

```
S4  assign d1 = sigin ^ q3;
S5  assign d2 = q1 ^ 3;
S6  assign d3 = q2 ^ q3;
S7  assign signout = q3;
```

Combinational component — 902

```
S8  // a clock generator
    always
    begin
       clk = 0;
       #5;
       clk = 1;
       #5;
    end
```

Clock component — 903

```
S9  // a signal generator
    always
    begin
       #10;
       sigin = $random;
    end
```

Test-bench component (Driver) — 904

```
    // initialization
    initial
    begin
S10    reset = 0;
       sigin = 0;
       #1;
S11    reset = 1;
       #5;
S12    $monitor($time, "%b, %b", sigin, sigout);
       #1000 $finish;
    end
    end module
```

Test-bench component (initialization) — 905

Test-bench component (monitor) — 906

FIG. 28

SIGNAL NETWORK ANALYSIS

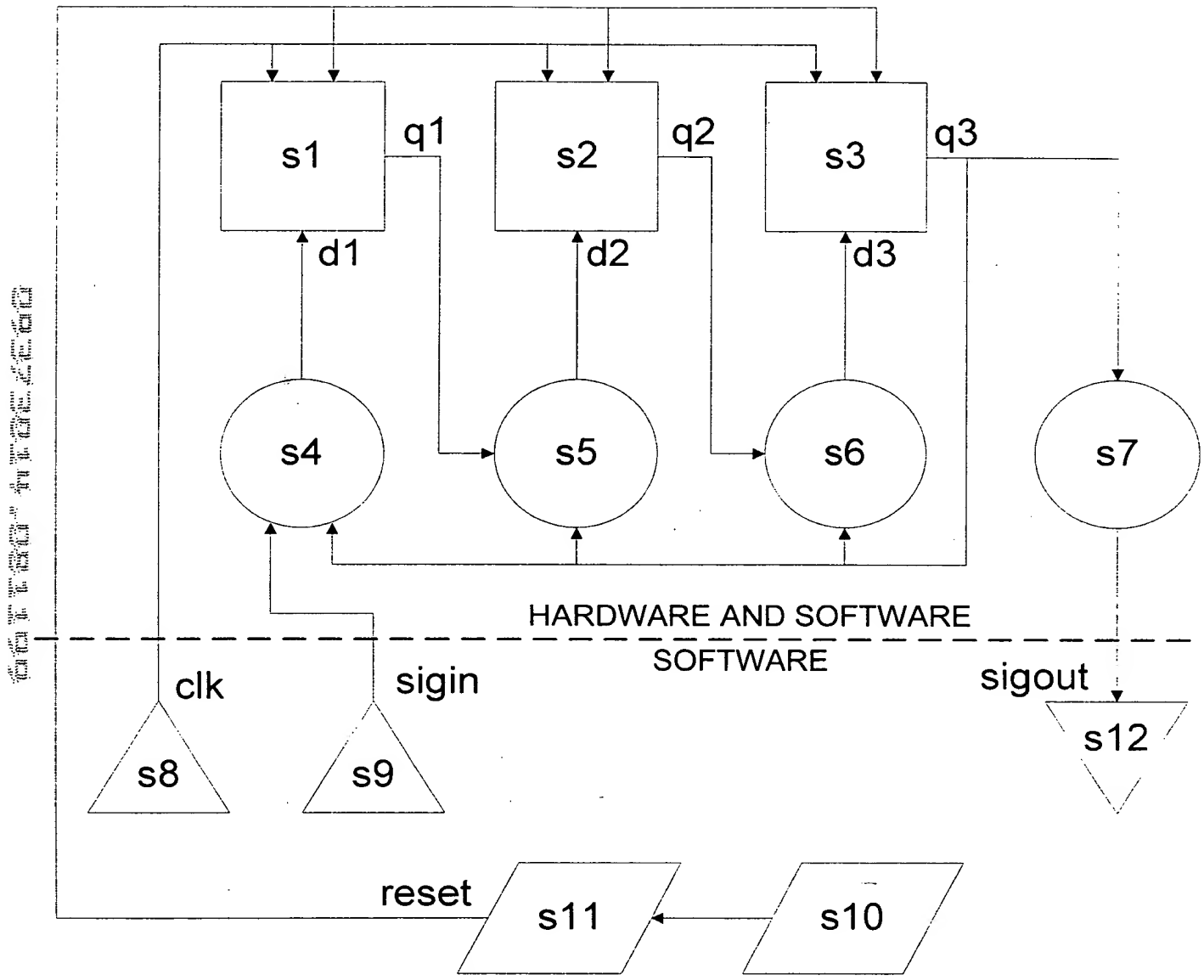


FIG. 29

SOFTWARE/HARDWARE PARTITION RESULT

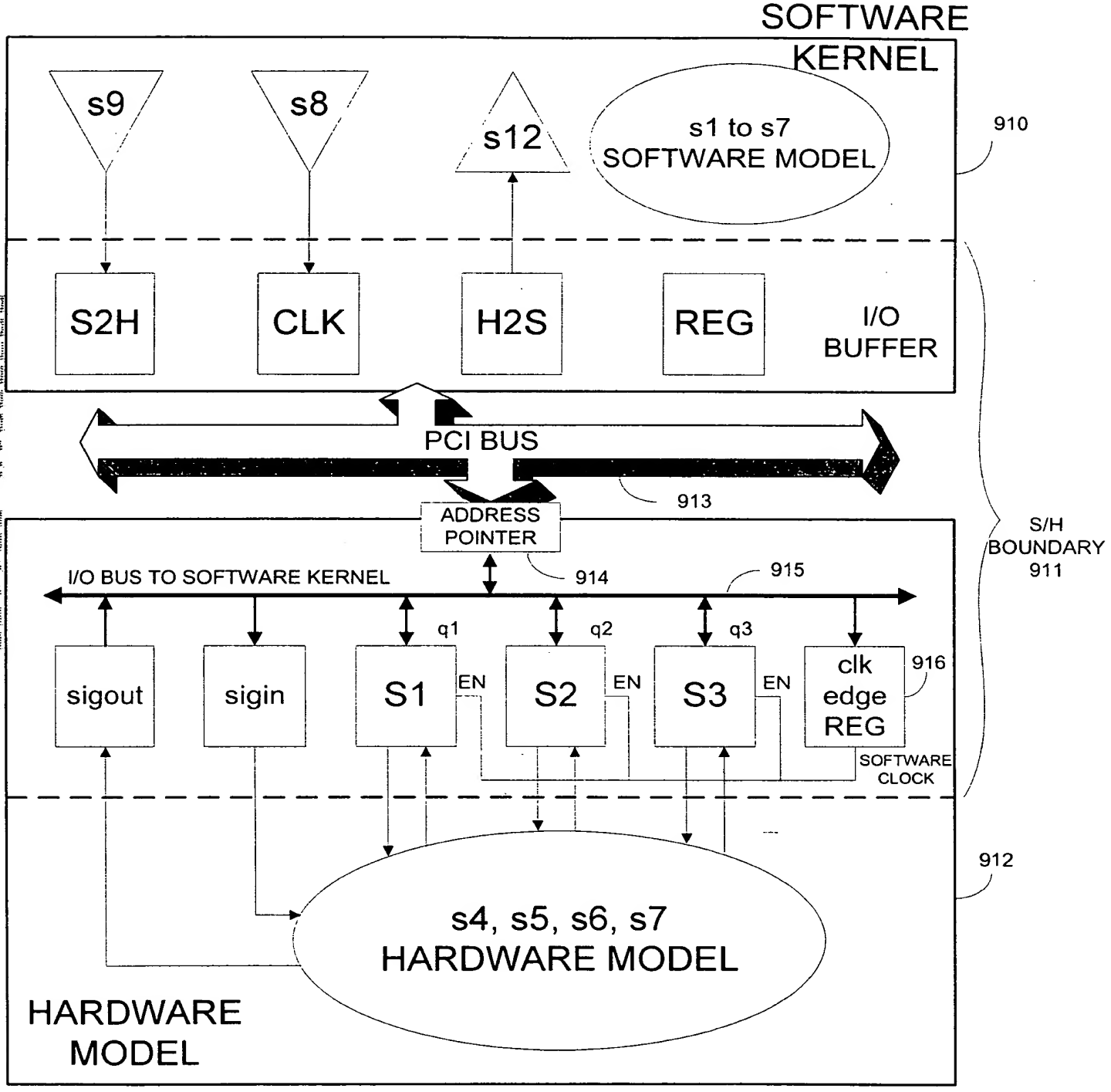


FIG. 30

HARDWARE MODEL

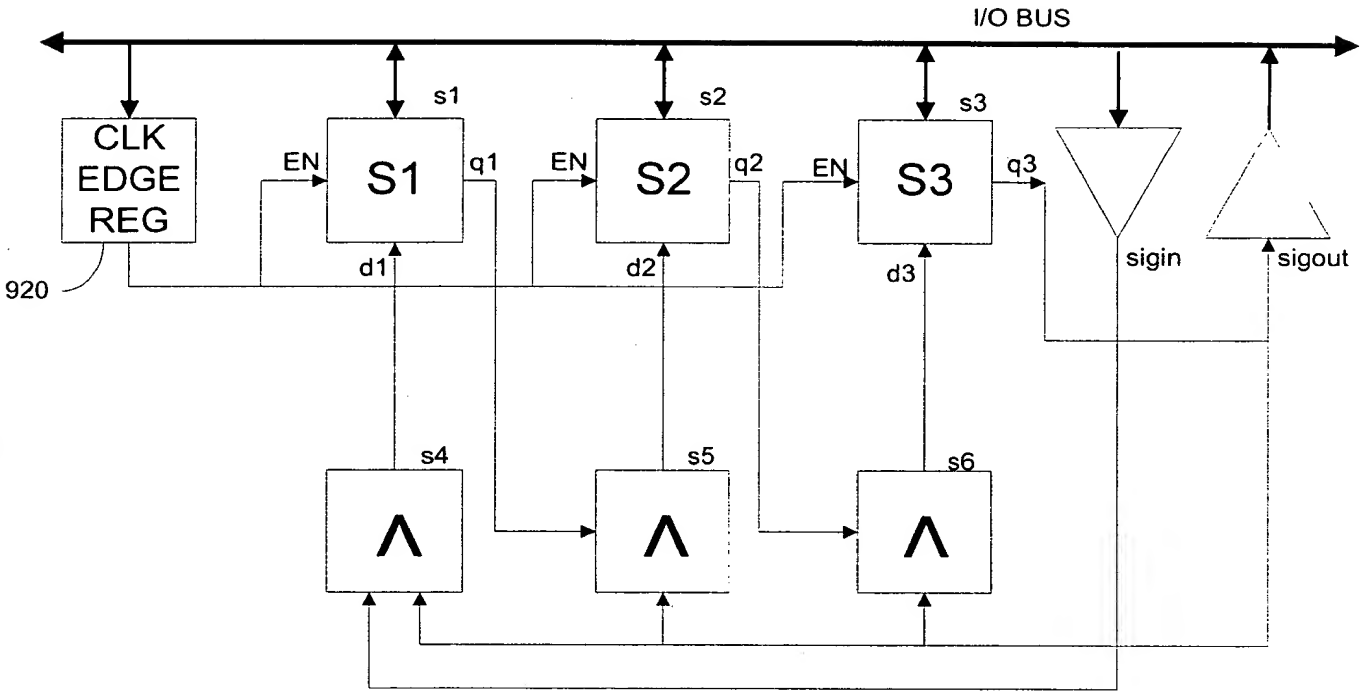
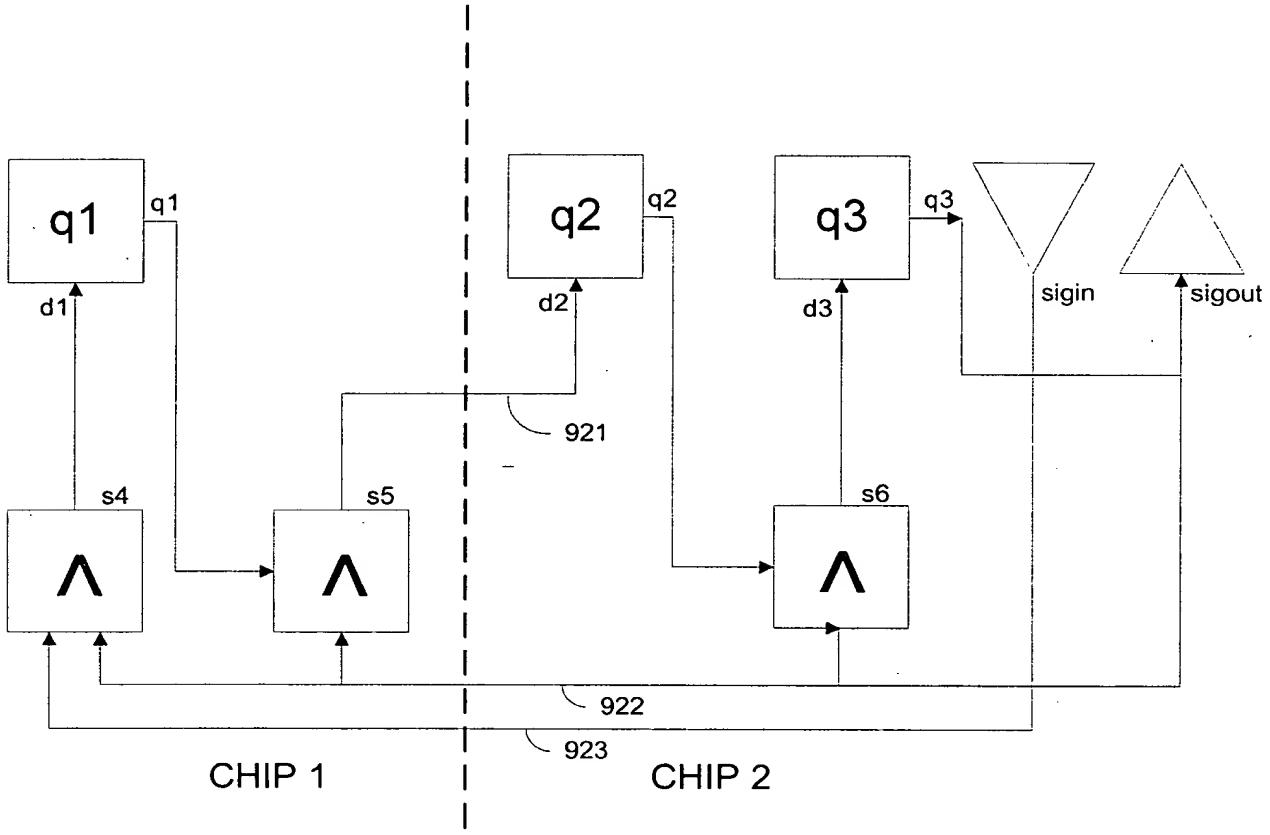


FIG. 31

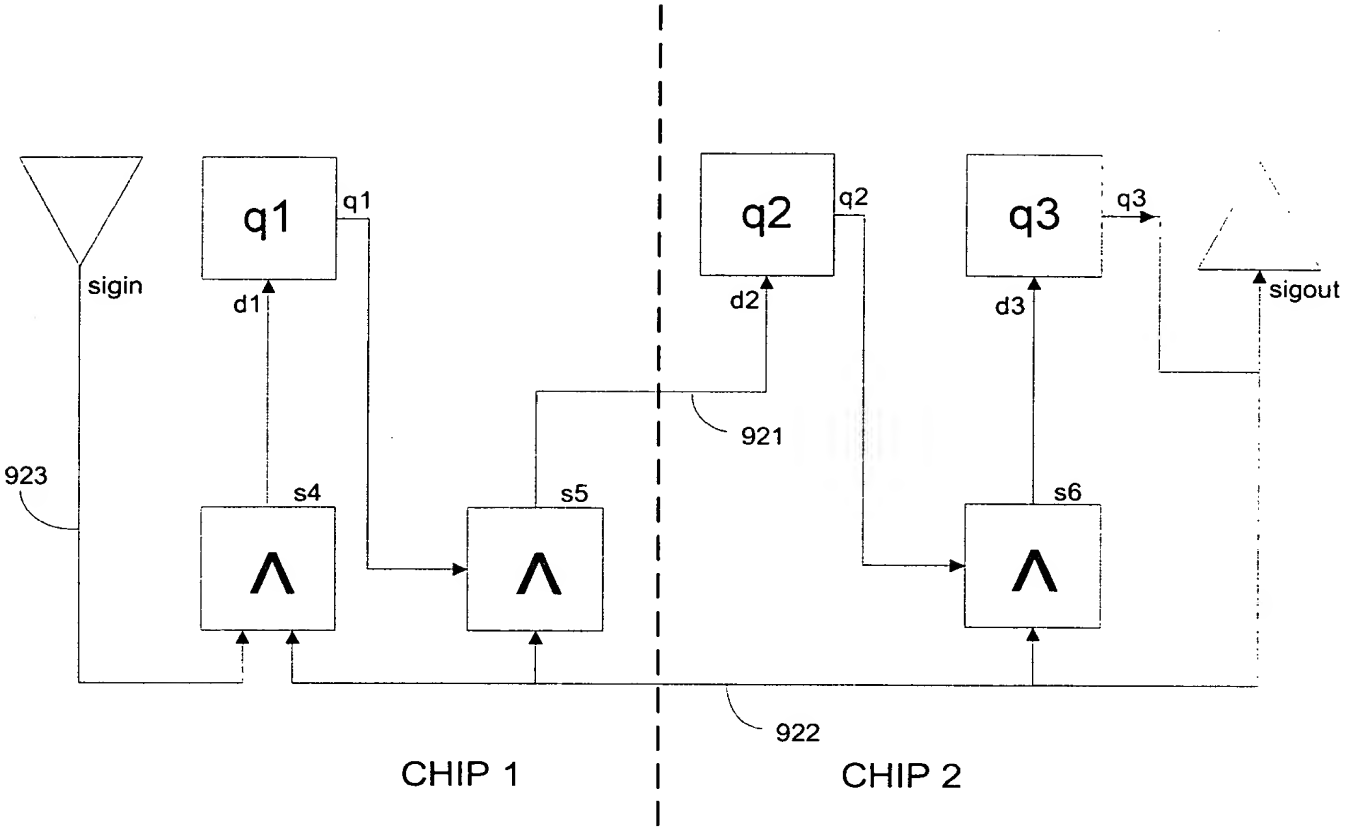
PARTITION RESULT #1



(IGNORE I/O AND CLOCK EDGE REGISTER)

FIG. 32

PARTITION RESULT #2



(IGNORE I/O AND CLOCK EDGE REGISTER).

FIG. 33

LOGIC PATCHING

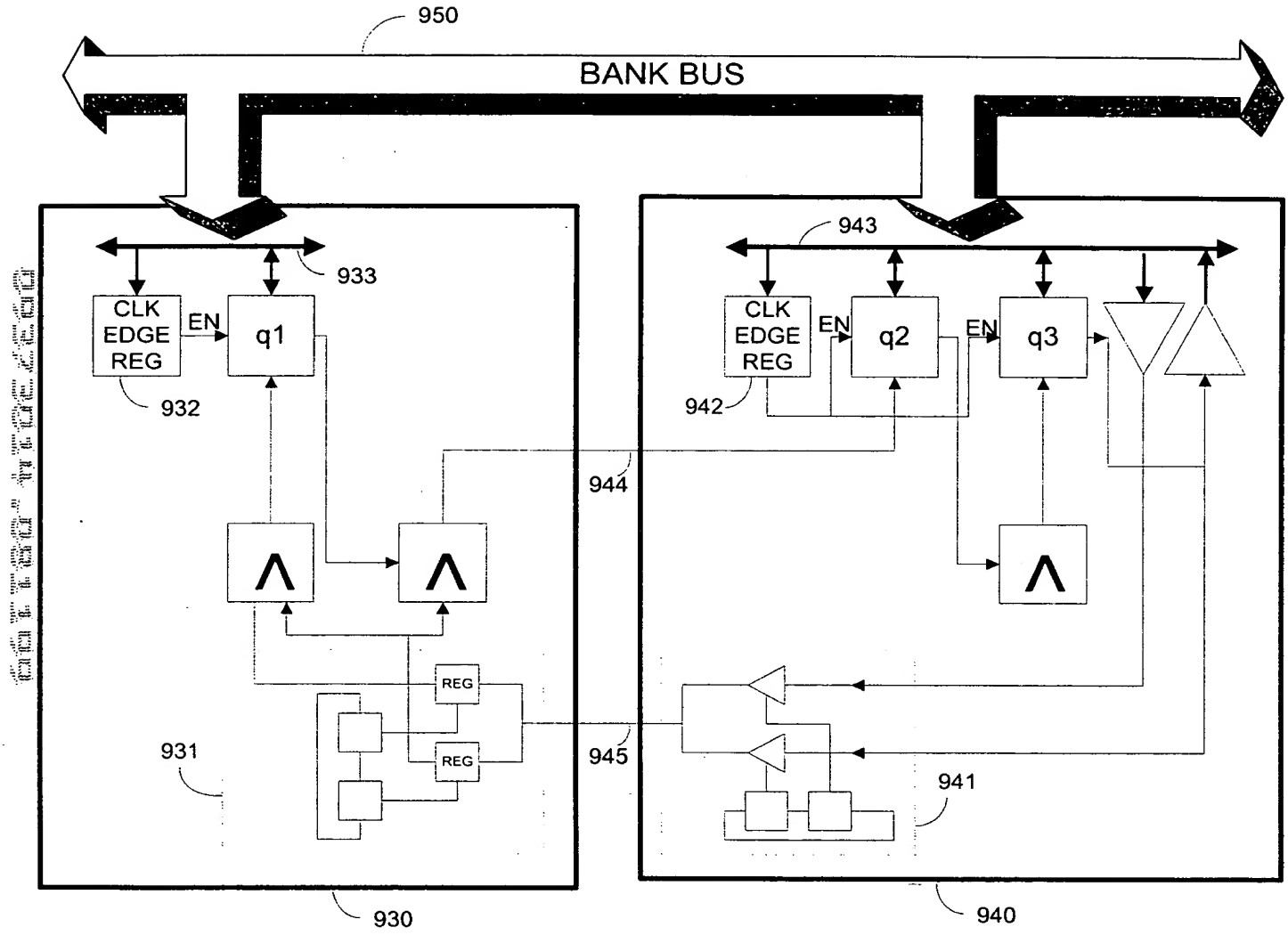
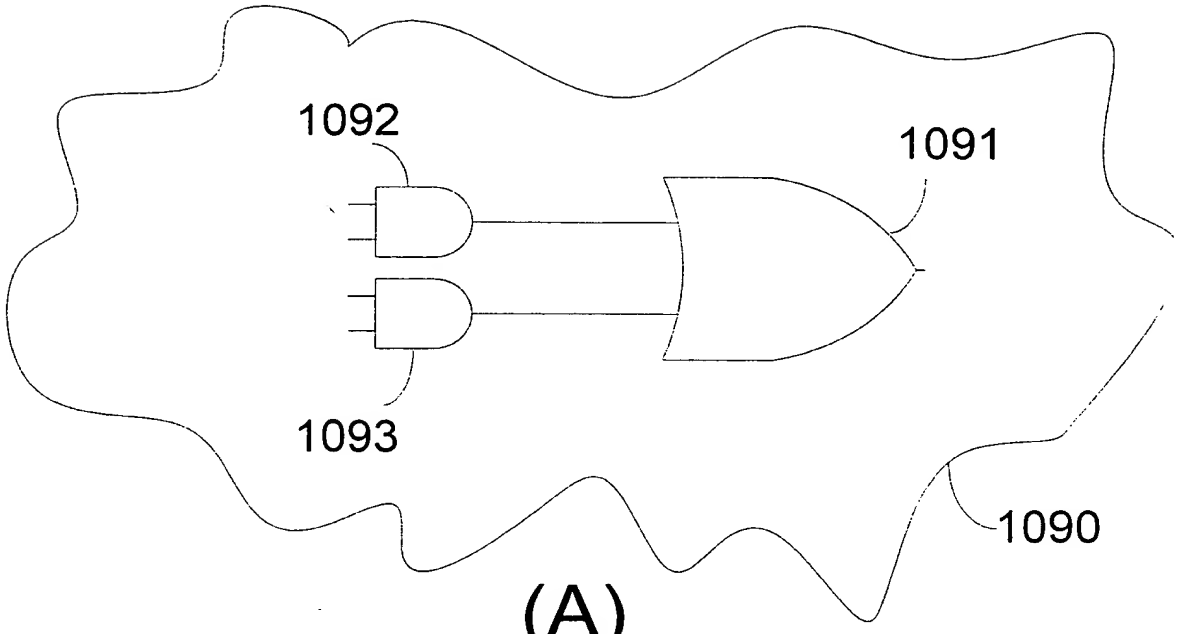
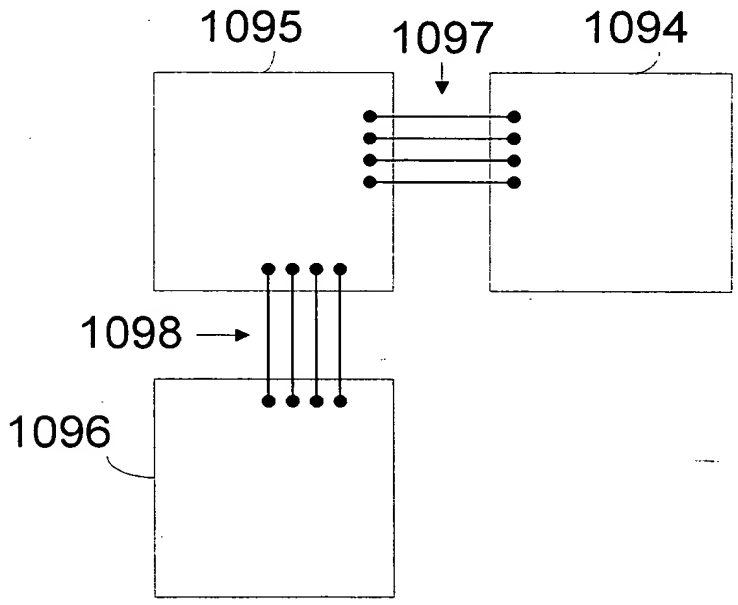


FIG. 34

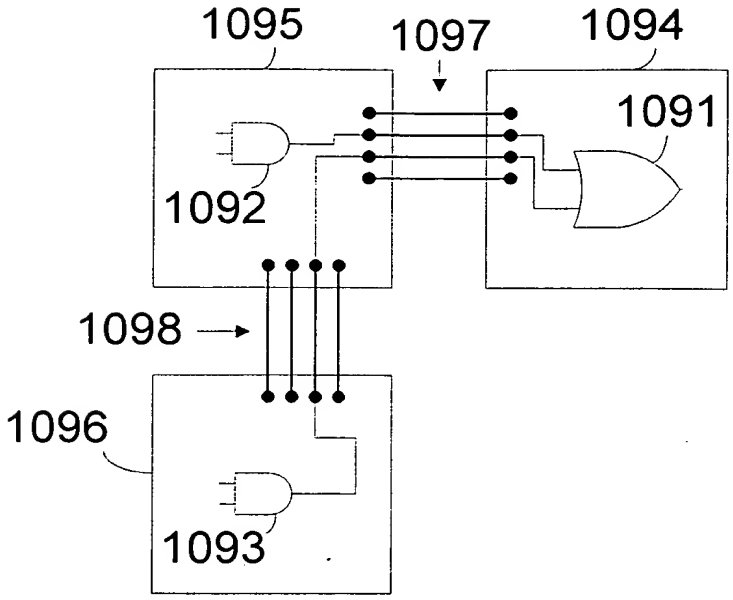


(A)

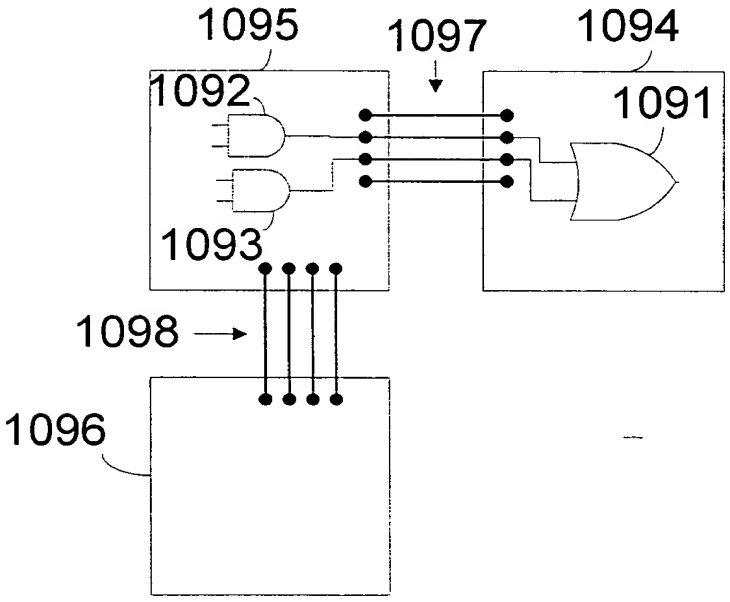


(B)

FIG. 35


1095
1097
1094
1091
1092
1098
1096
1093

(C)

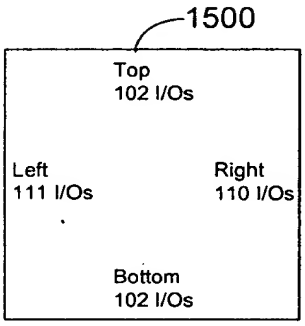

1095
1097
1094
1092
1091
1093
1098
1096

(D)

FIG. 35

I/O PIN OVERVIEW OF FPGA LOGIC DEVICE

FPGA : 10K130V, 10K250V with 599-pin PGA package



425 Interconnect I/O pins

45 Dedicated I/O pins:

GCLK, FD_BUS[31..0], F_RD, F_WR, DATAXSFR, SHIFTIN, SHIFTOUT, SPACE[2..0], EVAL, EV_REQ_N, DEV_OE, DEV_CLRN

FIG. 36

FPGA INTERCONNECT BUSES

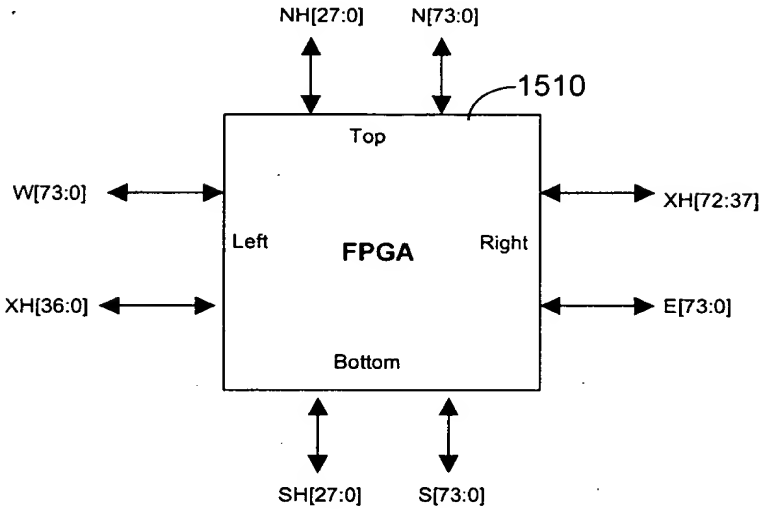


FIG. 37

BOARD CONNECTION - SIDE VIEW

DUAL-BOARD CONFIGURATION

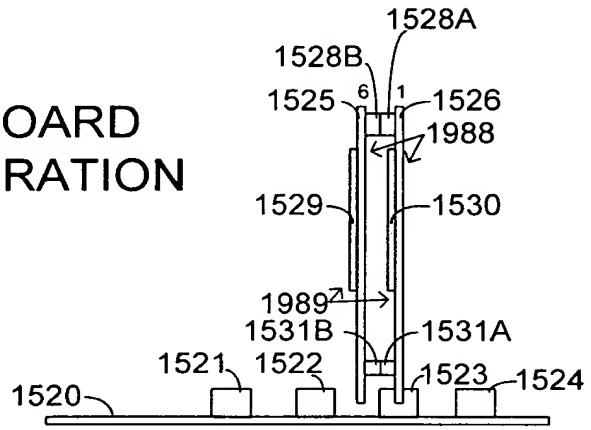


FIG. 38(A)

SIX BOARD CONFIGURATION

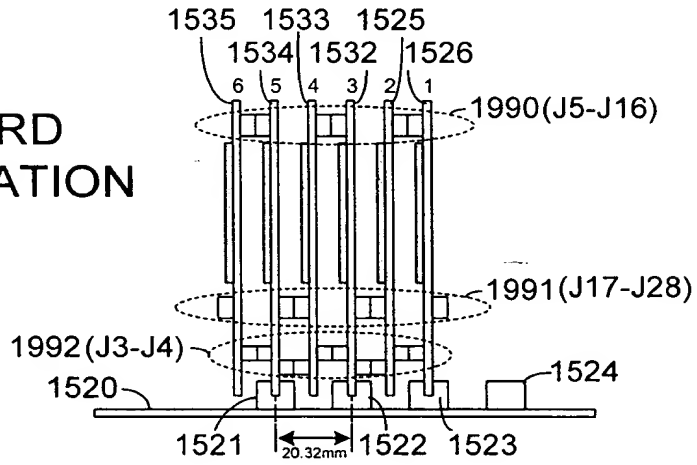


FIG. 38(B)

SIX-BOARD CONFIGURATION
DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY – X TORUS, Y MESH

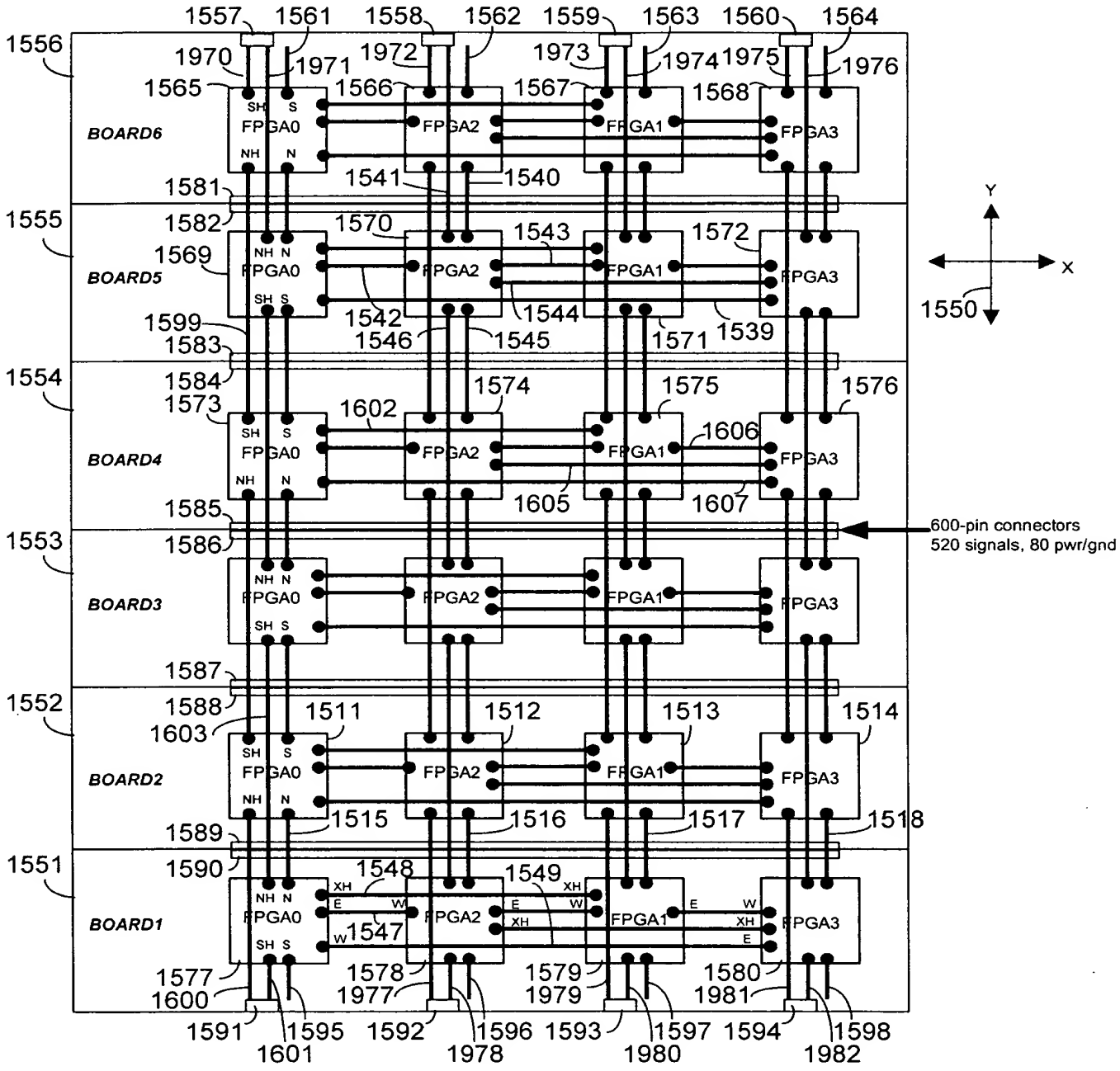


FIG. 39

FPGA ARRAY CONNECTION BETWEEN BOARDS

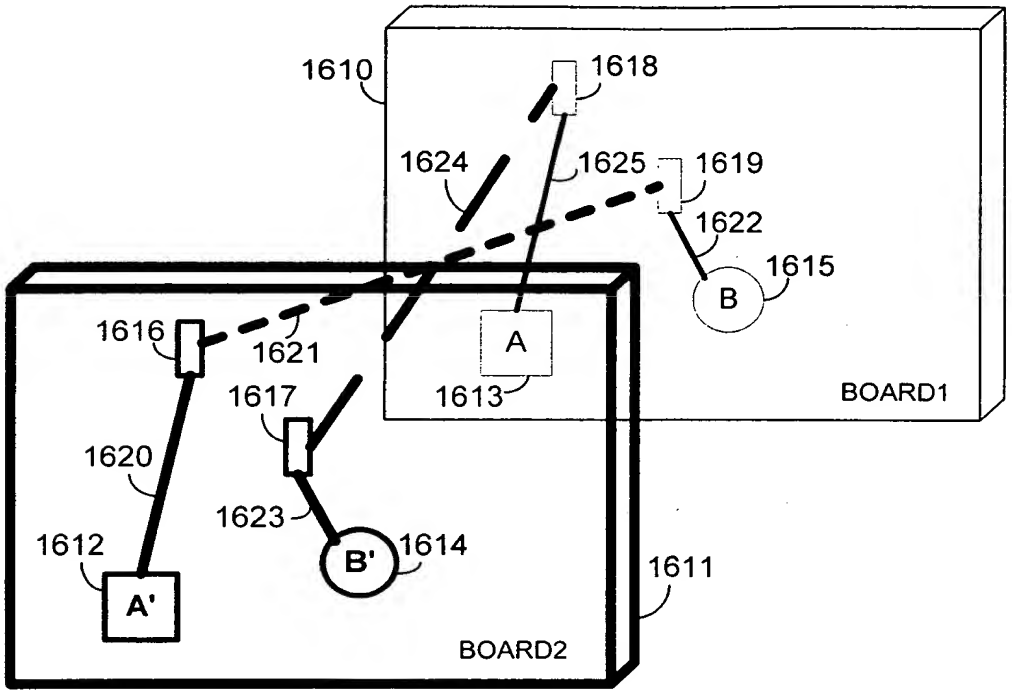


FIG. 40(A)

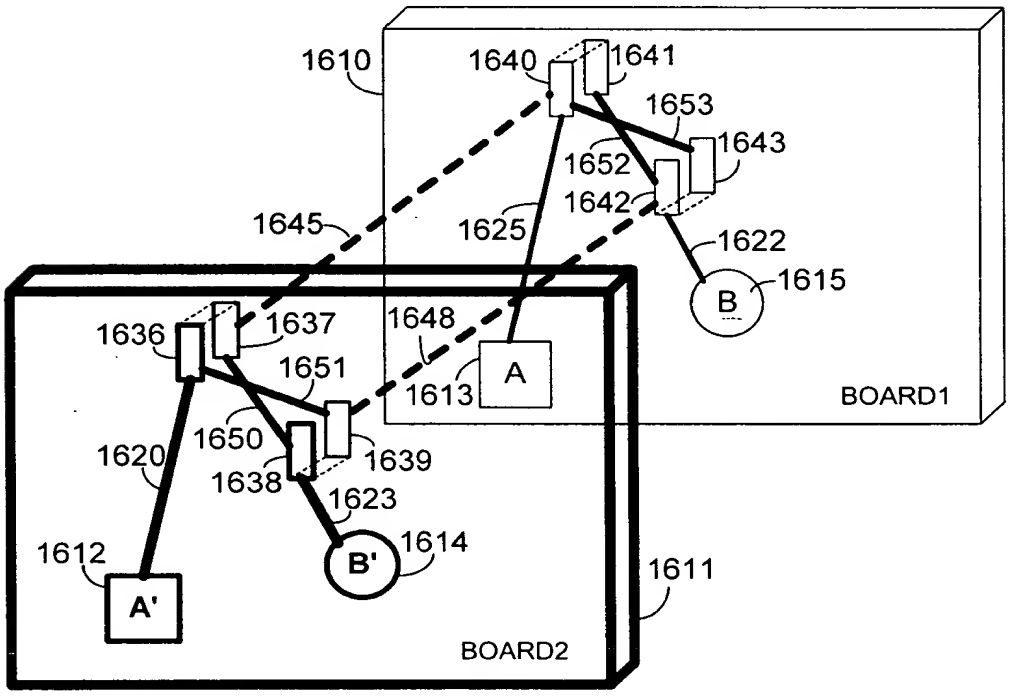


FIG. 40(B)

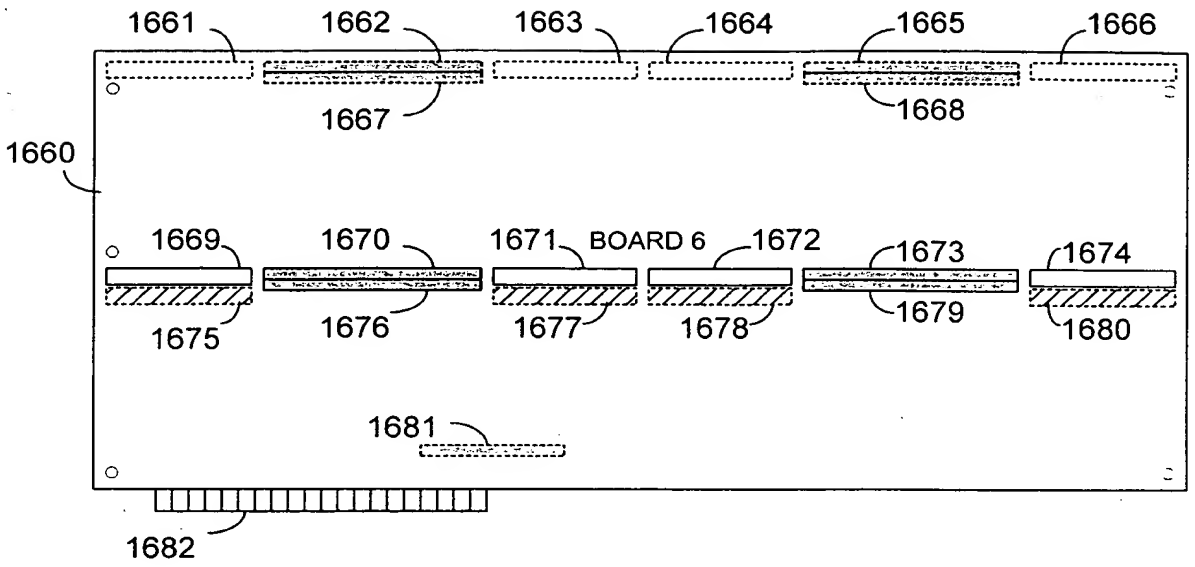


FIG. 41(A)

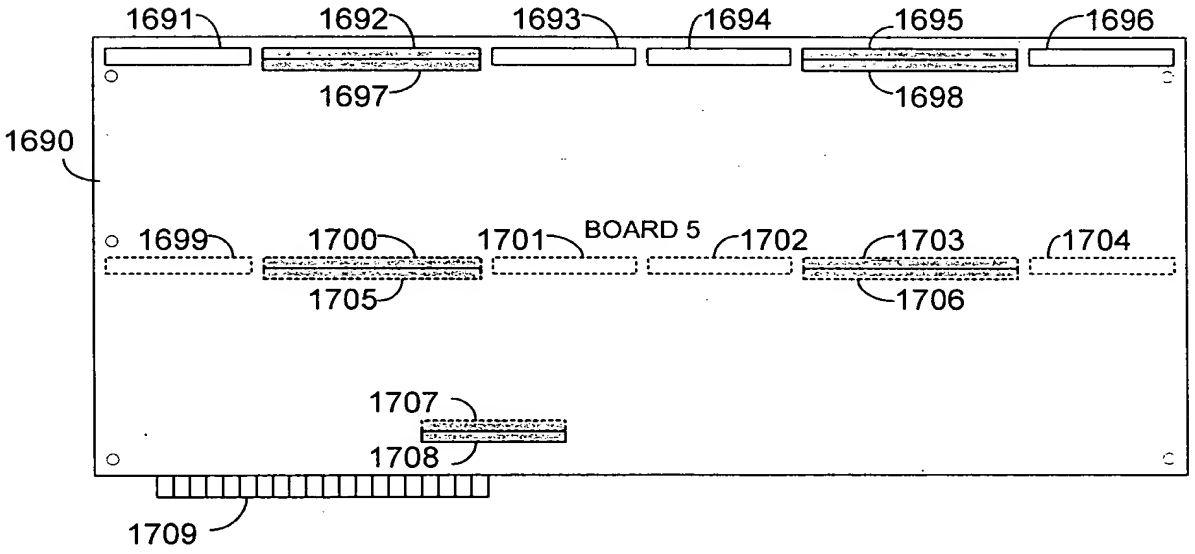


FIG. 41(B)

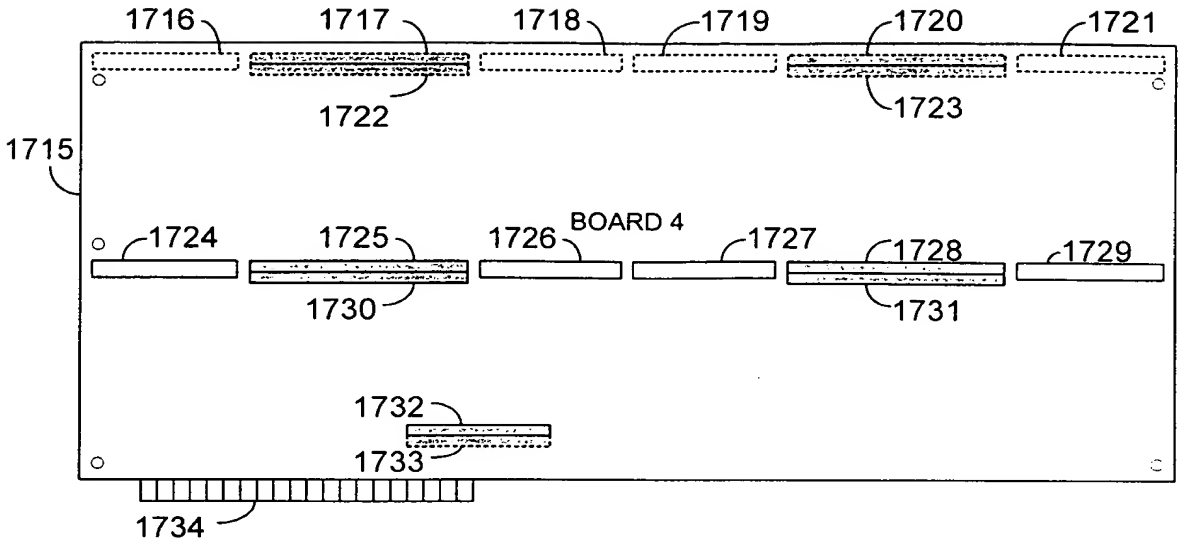


FIG. 41(C)

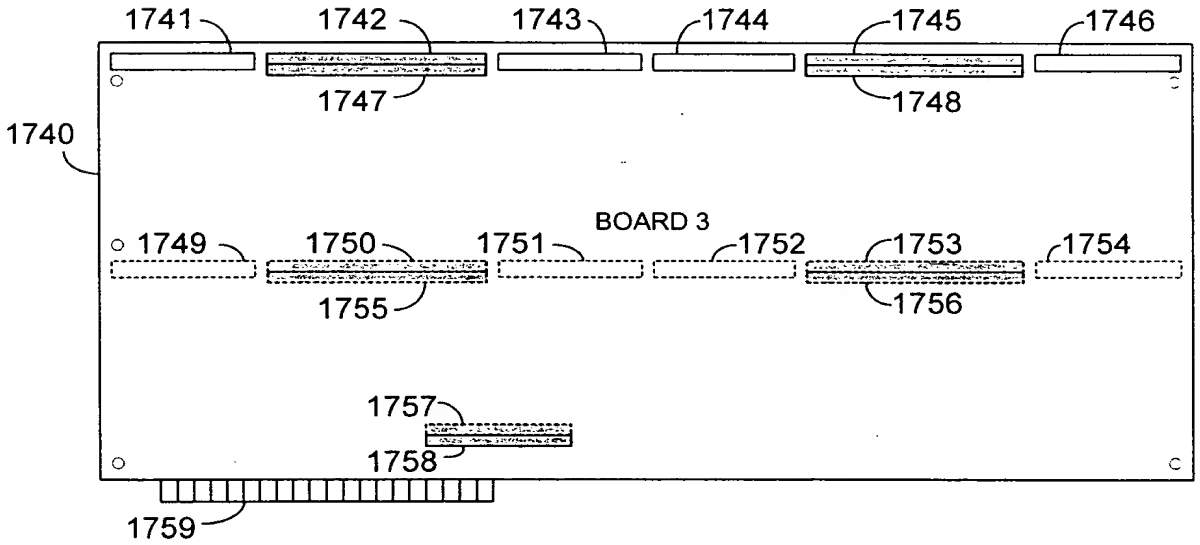


FIG. 41(D)

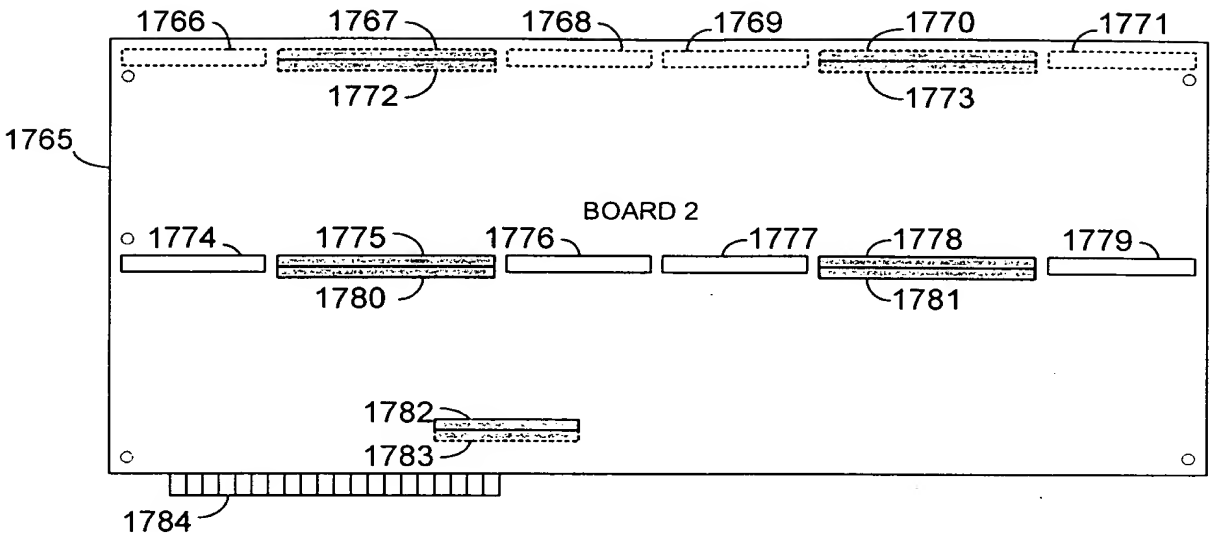


FIG. 41(E)

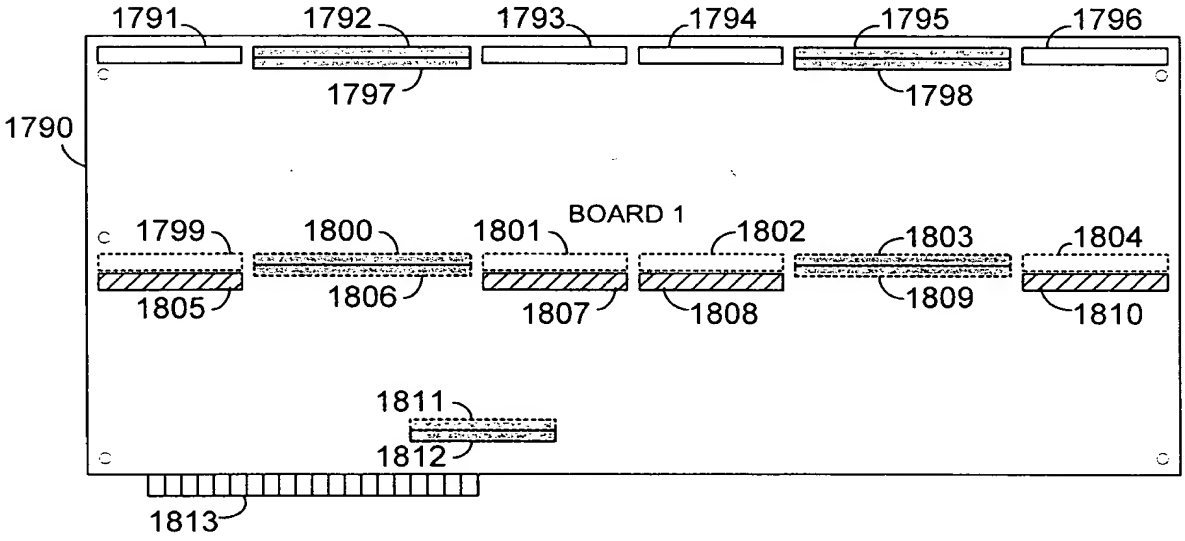


FIG. 41(F)

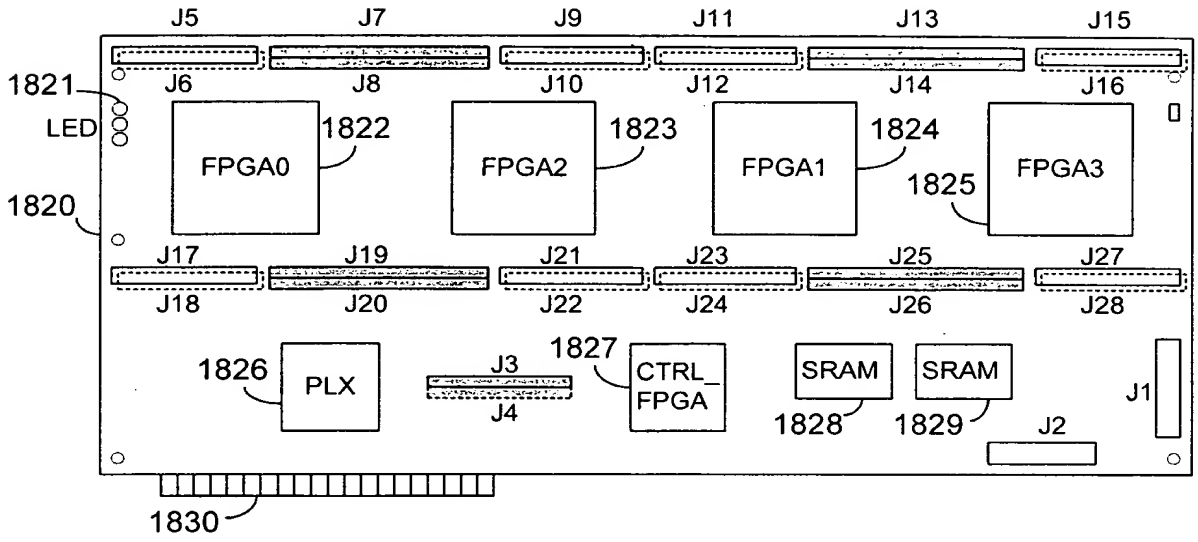

J5
J7
J9
J11
J13
J15
J6
J8
J10
J12
J14
J16
1821
LED
1820
FPGA0
1822
FPGA2
1823
FPGA1
1824
1825
FPGA3
J17
J19
J21
J23
J25
J27
J18
J20
J22
J24
J26
J28
1826
PLX
J3
J4
1827
CTRL_
FPGA
SRAM
SRAM
1828
1829
J1
J2
1830

FIG. 42

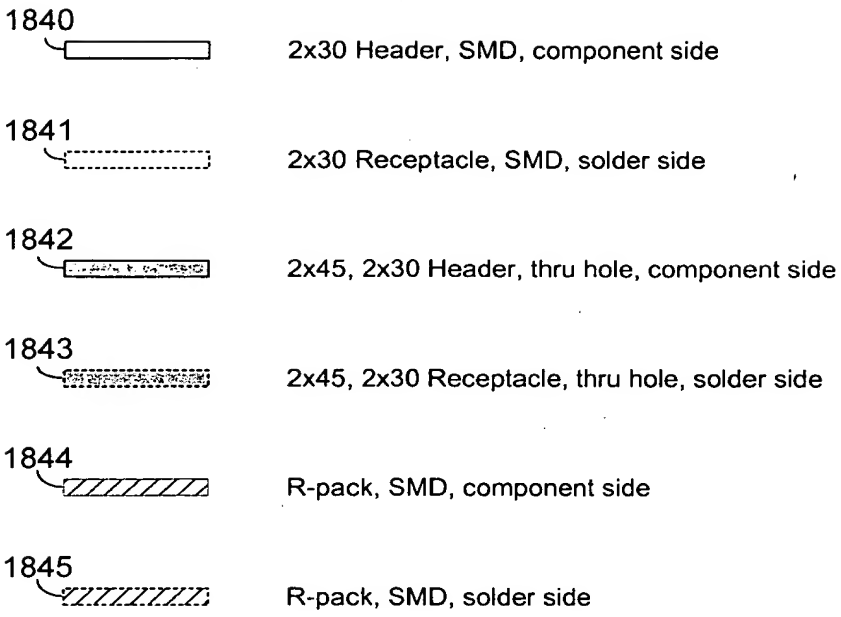

1840
2x30 Header, SMD, component side
1841
2x30 Receptacle, SMD, solder side
1842
2x45, 2x30 Header, thru hole, component side
1843
2x45, 2x30 Receptacle, thru hole, solder side
1844
R-pack, SMD, component side
1845
R-pack, SMD, solder side

FIG. 43

TWO-BOARD CONFIGURATION
DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY – X TORUS, Y MESH

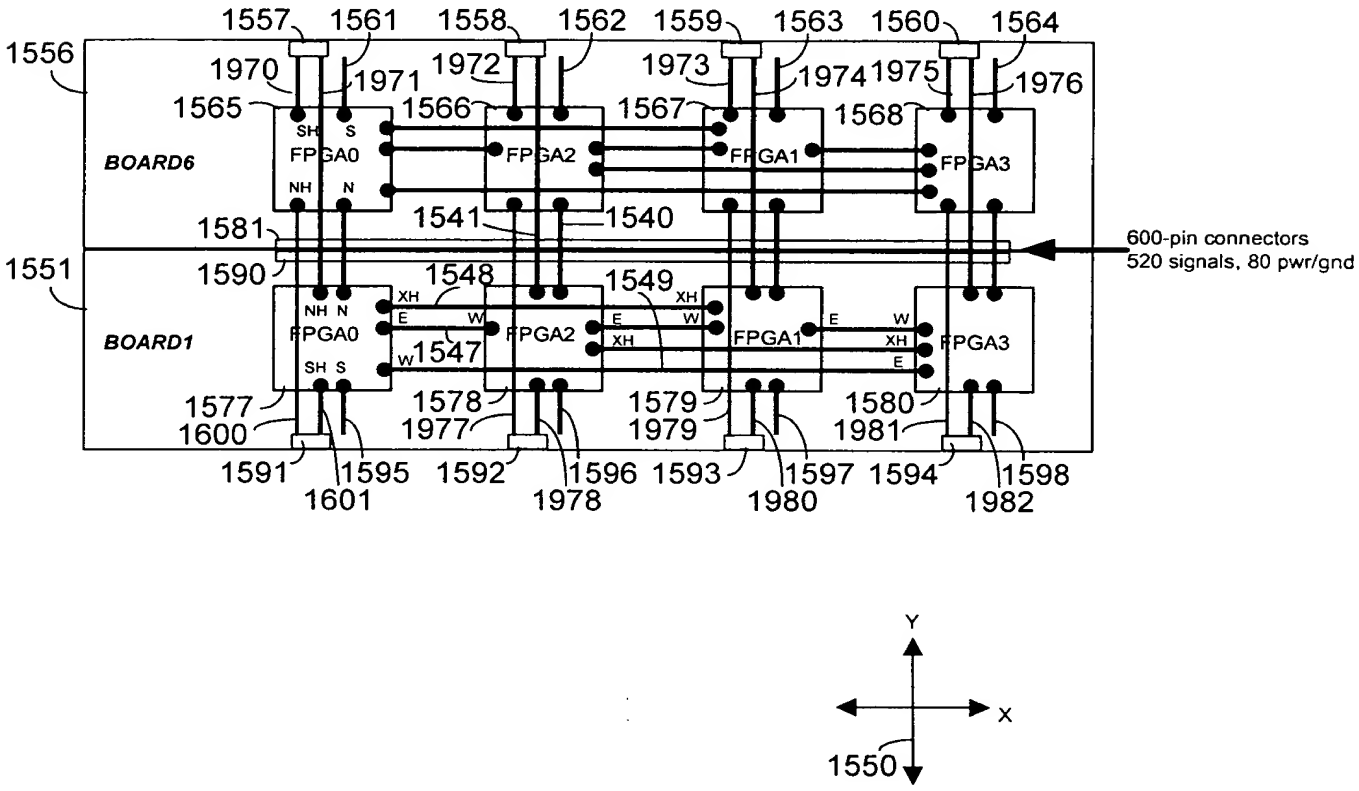


FIG. 44

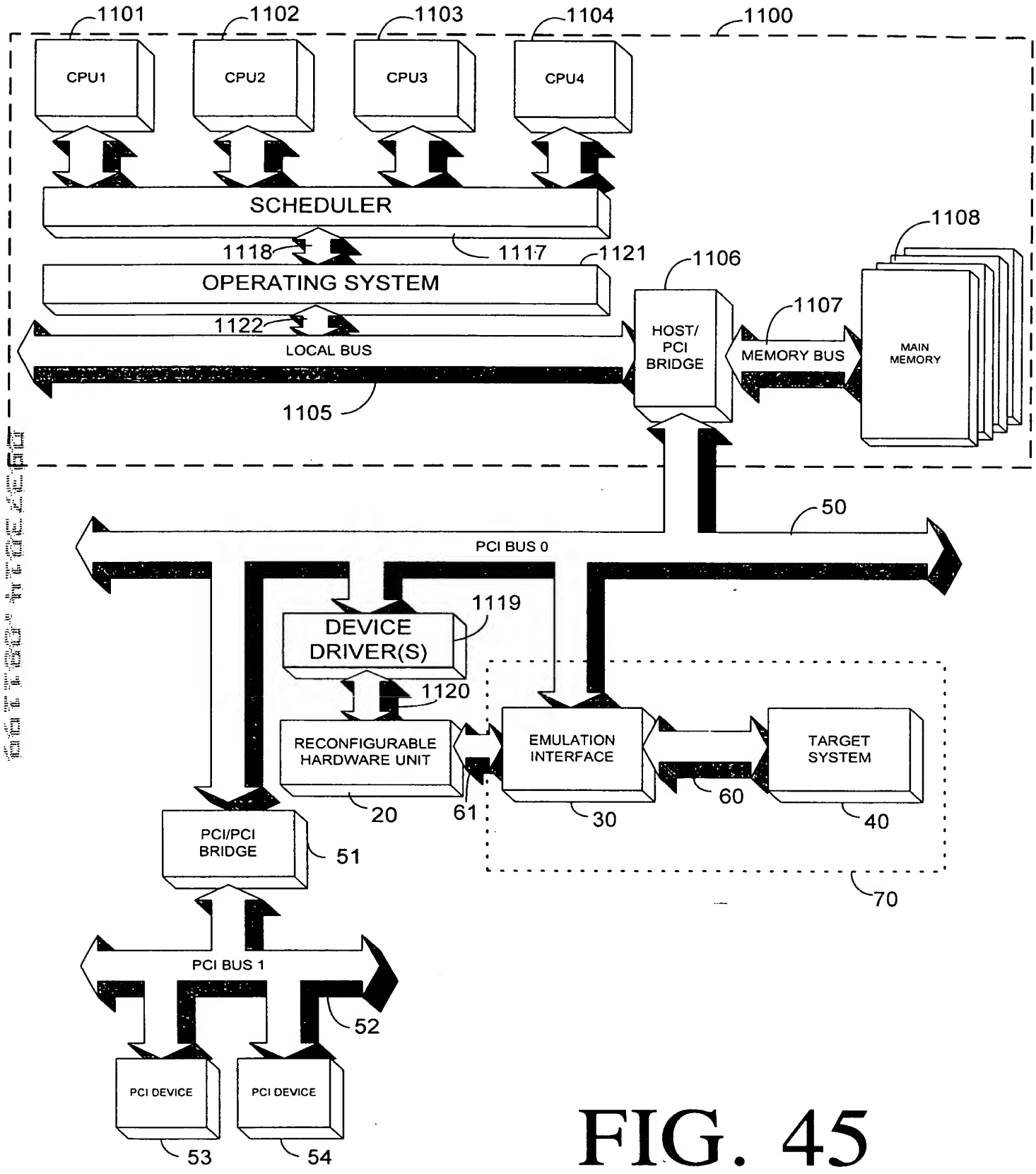

1101
1102
1103
1104
1100
CPU1
CPU2
CPU3
CPU4
SCHEDULER
1118
1117
1121
1106
1108
OPERATING SYSTEM
1122
1107
LOCAL BUS
HOST/
PCI
BRIDGE
MEMORY BUS
MAIN
MEMORY
1105
50
PCI BUS 0
1119
DEVICE
DRIVER(S)
1120
RECONFIGURABLE
HARDWARE UNIT
EMULATION
INTERFACE
TARGET
SYSTEM
20
61
30
60
40
PCI/PCI
BRIDGE
51
70
PCI BUS 1
52
PCI DEVICE
PCI DEVICE
53
54

FIG. 45

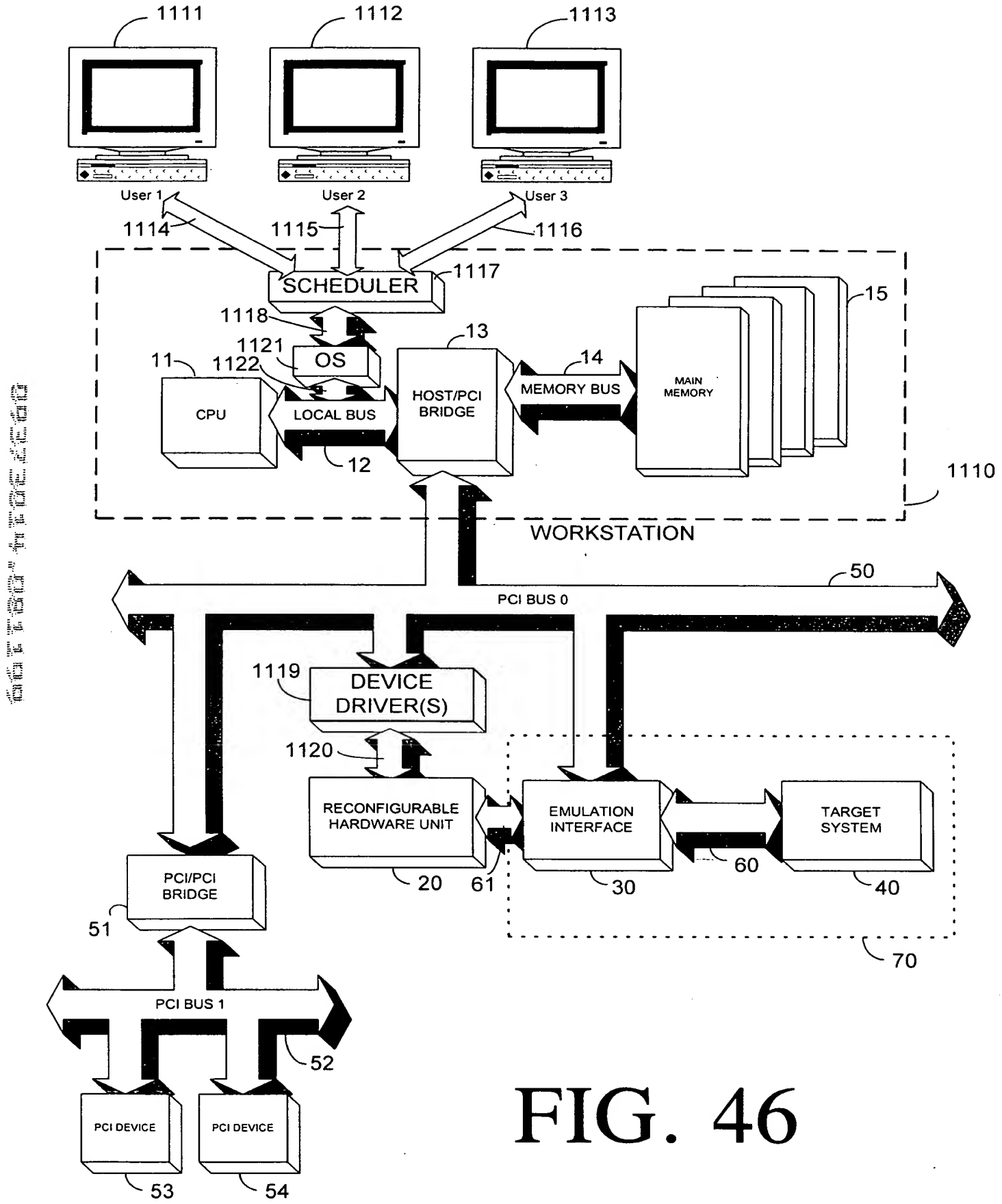

1111
1112
1113
User 1
User 2
User 3
1114
1115
1116
SCHEDULER
1117
1118
1121
OS
1122
11
CPU
LOCAL BUS
12
HOST/PCI BRIDGE
13
MEMORY BUS
14
MAIN MEMORY
15
1110
WORKSTATION
50
PCI BUS 0
1119
DEVICE DRIVER(S)
1120
RECONFIGURABLE HARDWARE UNIT
20
61
EMULATION INTERFACE
30
60
TARGET SYSTEM
40
70
PCI/PCI BRIDGE
51
PCI BUS 1
52
PCI DEVICE
PCI DEVICE
53
54

FIG. 46

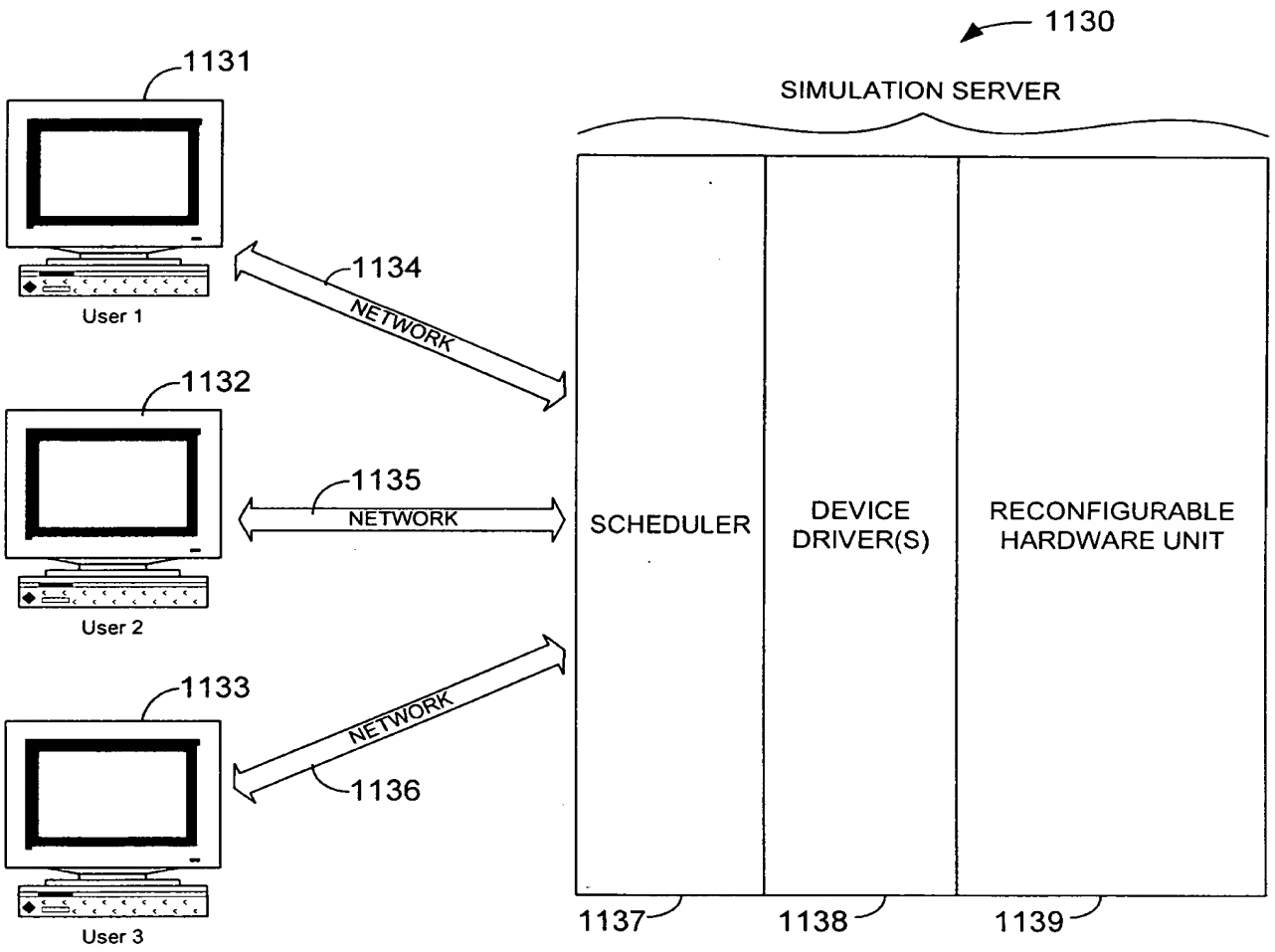

1130
SIMULATION SERVER
1131
User 1
1134
NETWORK
1132
User 2
1135
NETWORK
1133
User 3
NETWORK
1136
SCHEDULER
DEVICE DRIVER(S)
RECONFIGURABLE HARDWARE UNIT
1137
1138
1139

FIG. 47

SIMULATION SERVER ARCHITECTURE

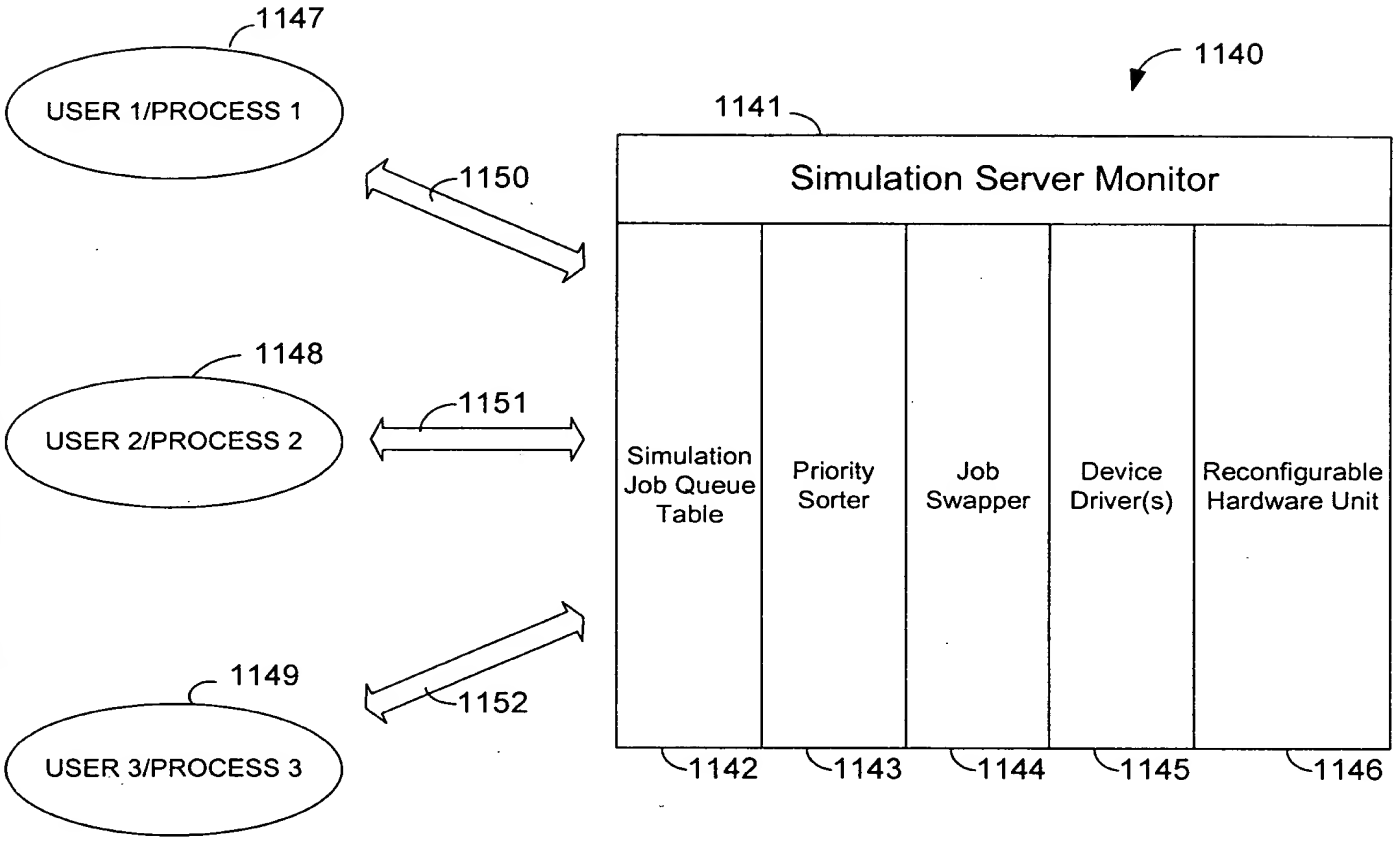


FIG. 48

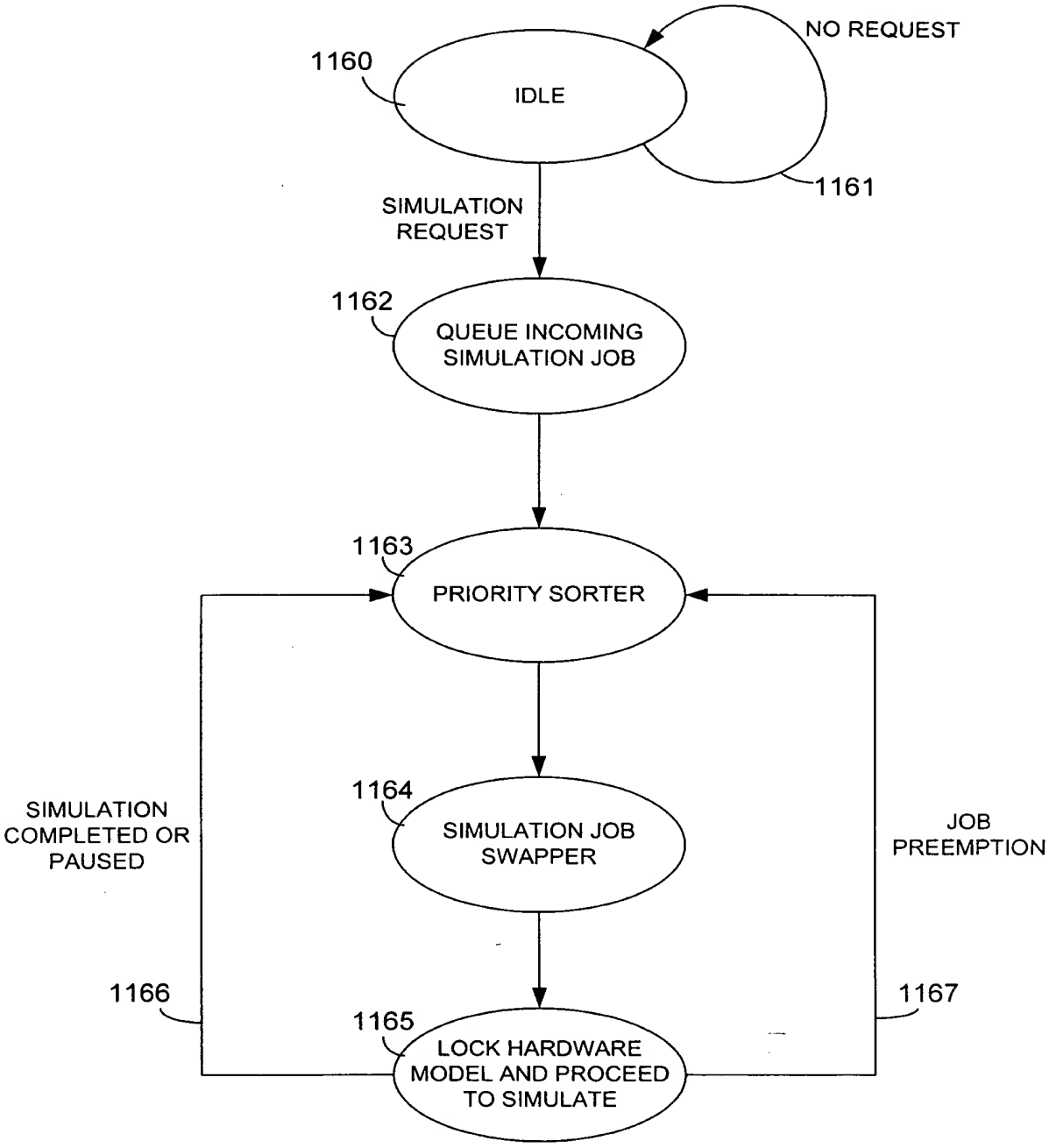

NO REQUEST
1160
IDLE
1161
SIMULATION REQUEST
1162
QUEUE INCOMING SIMULATION JOB
1163
PRIORITY SORTER
1164
SIMULATION JOB SWAPPER
SIMULATION COMPLETED OR PAUSED
JOB PREEMPTION
1166
1167
1165
LOCK HARDWARE MODEL AND PROCEED TO SIMULATE

FIG. 49

JOB SWAPPER

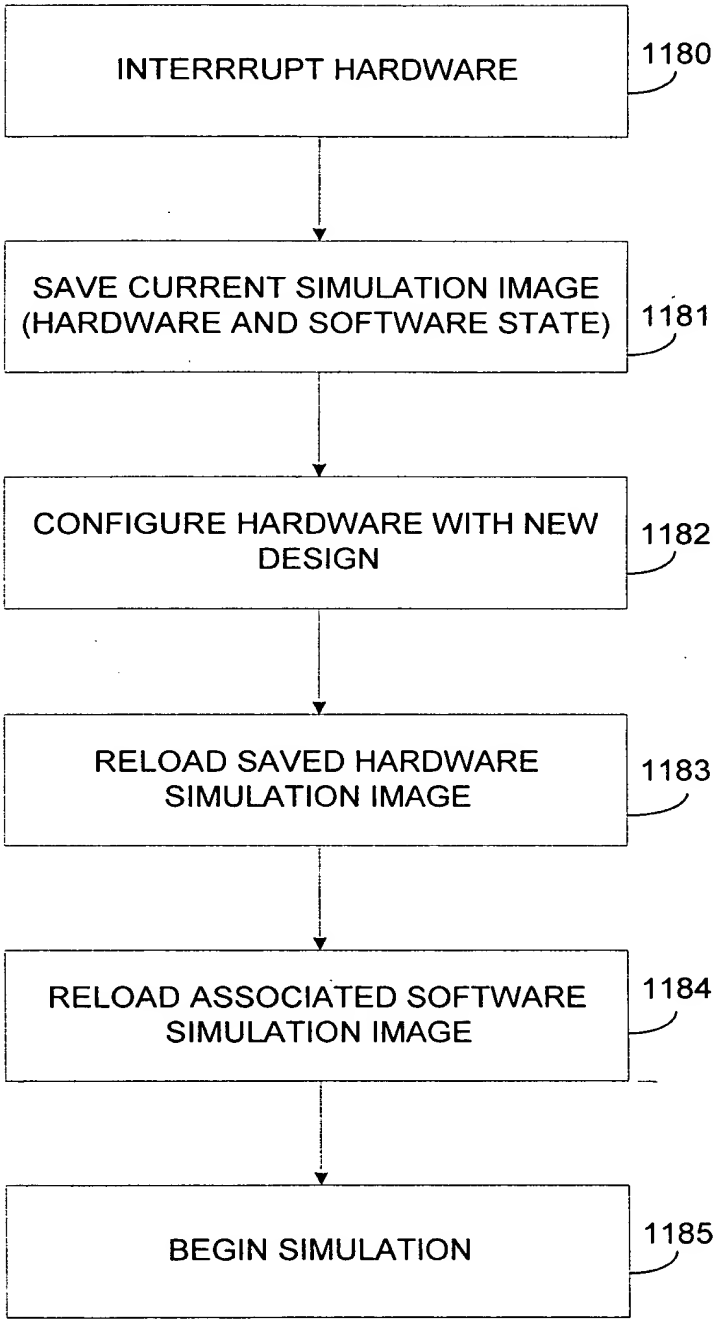


FIG. 50

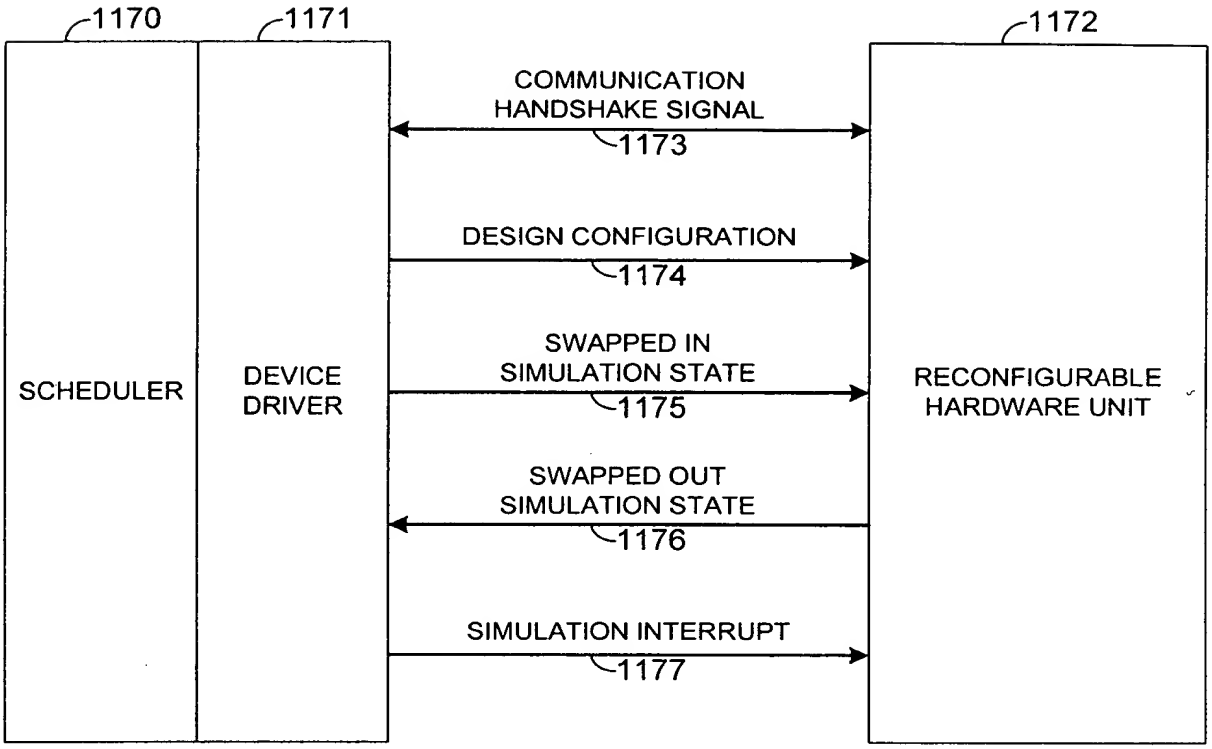

1170
1171
1172
SCHEDULER
DEVICE DRIVER
RECONFIGURABLE HARDWARE UNIT
COMMUNICATION HANDSHAKE SIGNAL
1173
DESIGN CONFIGURATION
1174
SWAPPED IN SIMULATION STATE
1175
SWAPPED OUT SIMULATION STATE
1176
SIMULATION INTERRUPT
1177

FIG. 51

PRIORITY I { JOB A, JOB B }

PRIORITY II { JOB C, JOB D }

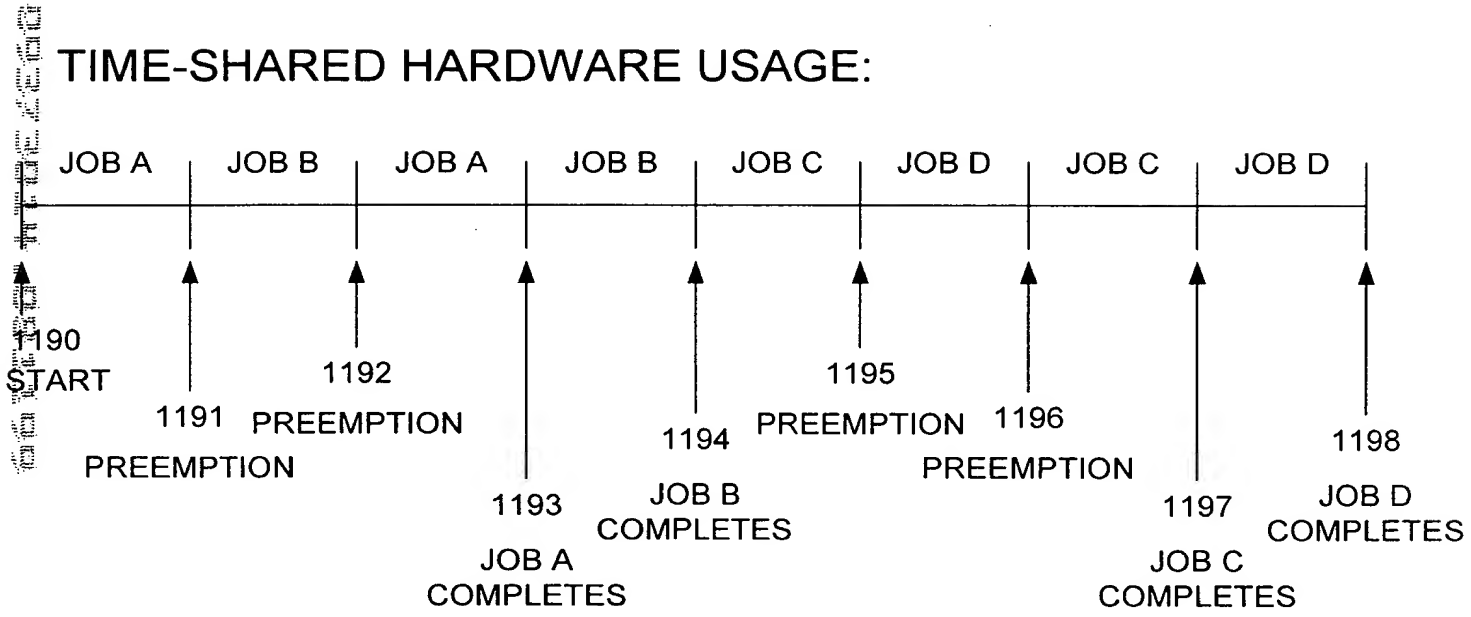


FIG. 52

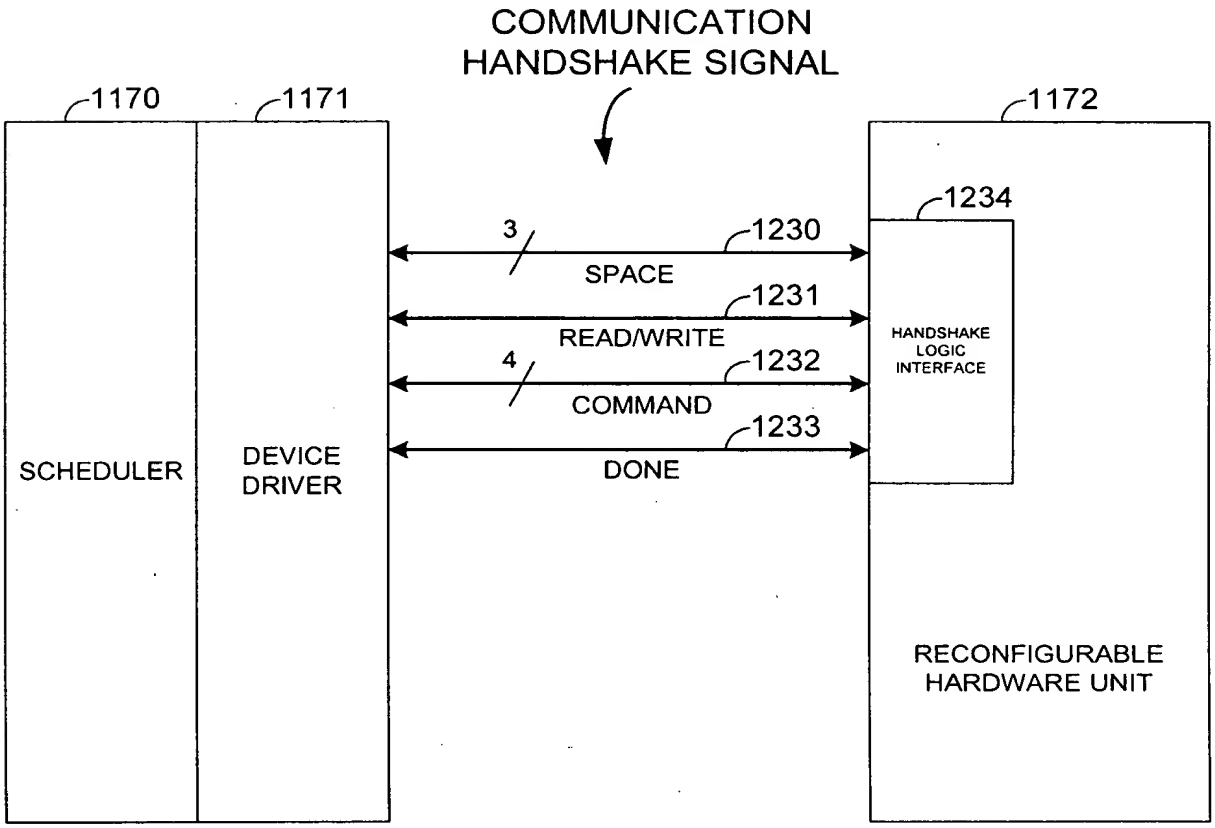

COMMUNICATION
HANDSHAKE SIGNAL
1170
1171
1172
1234
3
1230
SPACE
1231
READ/WRITE
4
1232
COMMAND
1233
DONE
HANDSHAKE
LOGIC
INTERFACE
SCHEDULER
DEVICE
DRIVER
RECONFIGURABLE
HARDWARE UNIT

FIG. 53

COMMUNICATION HANDSHAKE PROTOCOL

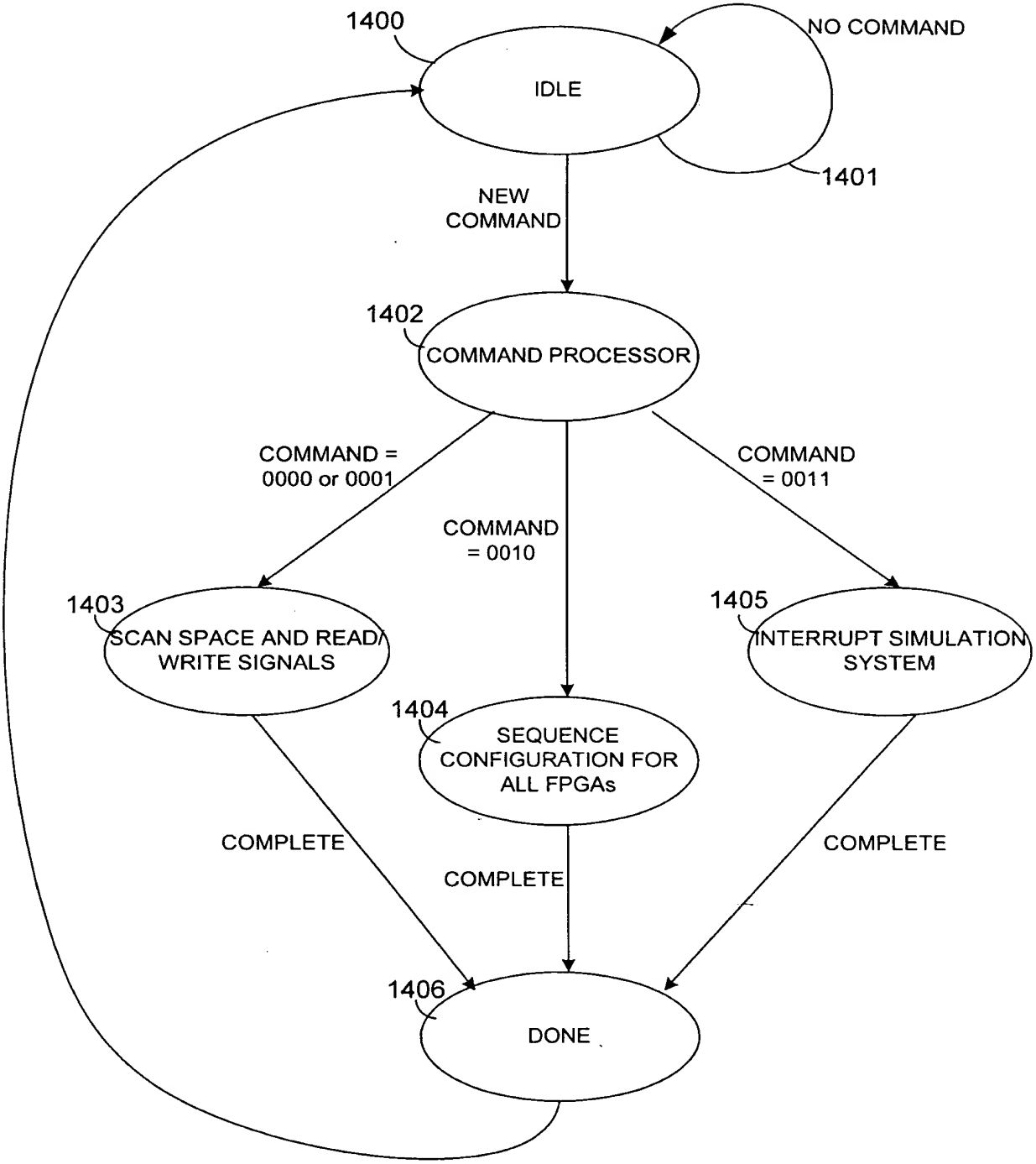


FIG. 54

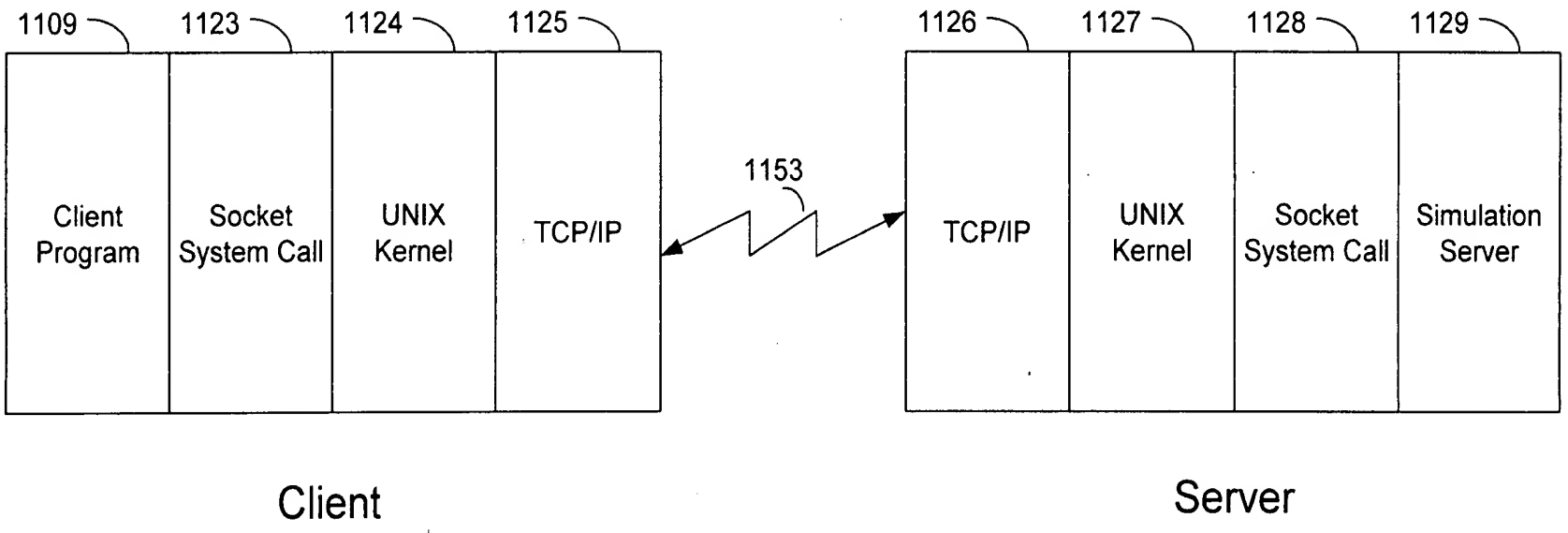

1109
1123
1124
1125
1126
1127
1128
1129
1153
Client Program
Socket System Call
UNIX Kernel
TCP/IP
TCP/IP
UNIX Kernel
Socket System Call
Simulation Server
Client
Server

FIG. 55

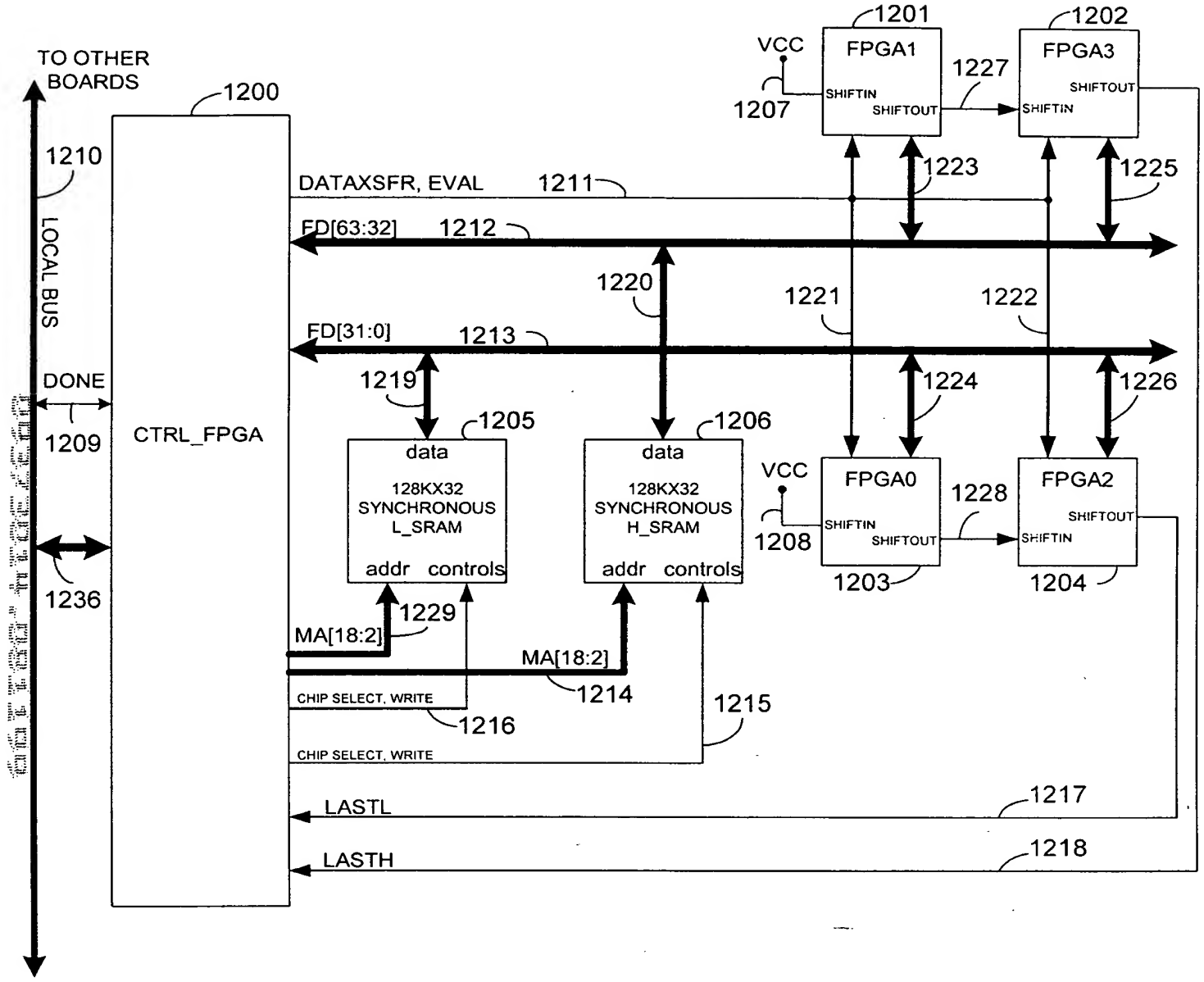

TO OTHER BOARDS
1200
1210
LOCAL BUS
DONE
1209
CTRL_FPGA
1236
DATAXSFR, EVAL
1211
FD[63:32]
1212
FD[31:0]
1213
1219
1205
data
128KX32
SYNCHRONOUS
L_SRAM
addr controls
1206
data
128KX32
SYNCHRONOUS
H_SRAM
addr controls
1229
MA[18:2]
MA[18:2]
1214
CHIP SELECT, WRITE
1216
CHIP SELECT, WRITE
1215
LASTL
1217
LASTH
1218
VCC
1207
1201
FPGA1
SHIFTIN
SHIFTOUT
1227
1202
FPGA3
SHIFTOUT
SHIFTIN
1223
1225
1220
1221
1222
1224
1226
VCC
1208
FPGA0
SHIFTIN
SHIFTOUT
1228
FPGA2
SHIFTOUT
SHIFTIN
1203
1204

FIG. 56

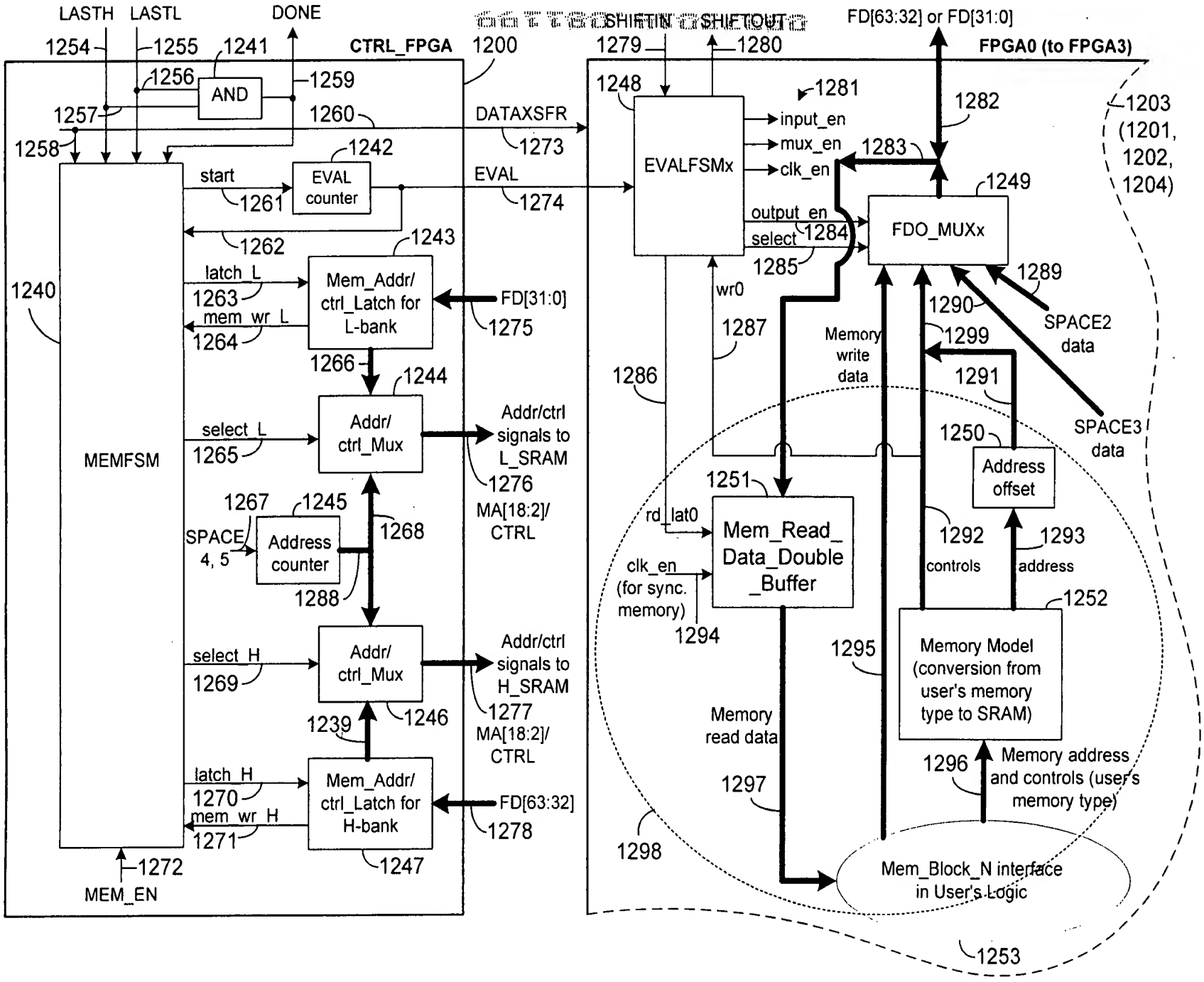

LASTH
LASTL
DONE
SHIFTIN
SHIFTOUT
FD[63:32] or FD[31:0]
1254
1255
1241
CTRL_FPGA
1200
1279
1280
FPGA0 (to FPGA3)
1256
AND
1259
1248
1281
1257
1260
DATAXSFR
input_en
1282
1203
(1201,
1202,
1204)
1258
1242
1273
mux_en
1283
start
EVAL
counter
EVAL
clk_en
EVALFSMx
1261
1274
1249
output_en
FDO_MUXx
1262
1243
select
1284
1285
1289
latch_L
Mem_Addr/
ctrl_Latch for
L-bank
FD[31:0]
wr0
1240
1263
1275
1290
SPACE2
data
mem_wr_L
1264
1287
Memory
write
data
1299
1266
1244
1286
1291
Addr/
ctrl_Mux
Addr/ctrl
signals to
L_SRAM
SPACE3
data
select_L
1250
MEMFSM
1265
1276
Address
offset
1251
1267
1245
MA[18:2]/
CTRL
rd_lat0
Mem_Read_
Data_Double
_Buffer
1292
1293
SPACE
4, 5
Address
counter
1268
clk_en
(for sync.
memory)
controls
address
1288
1294
1252
select_H
Addr/
ctrl_Mux
Addr/ctrl
signals to
H_SRAM
1295
Memory Model
(conversion from
user's memory
type to SRAM)
1269
1277
1246
1239
MA[18:2]/
CTRL
Memory
read data
latch_H
Mem_Addr/
ctrl_Latch for
H-bank
FD[63:32]
1296
Memory address
and controls (user's
memory type)
1270
1297
mem_wr_H
1271
1278
1298
1272
1247
MEM_EN
Mem_Block_N interface
in User's Logic
1253

FIG. 57

MEMFSM - Memory Finite State Machine in CTRL_FPGA unit

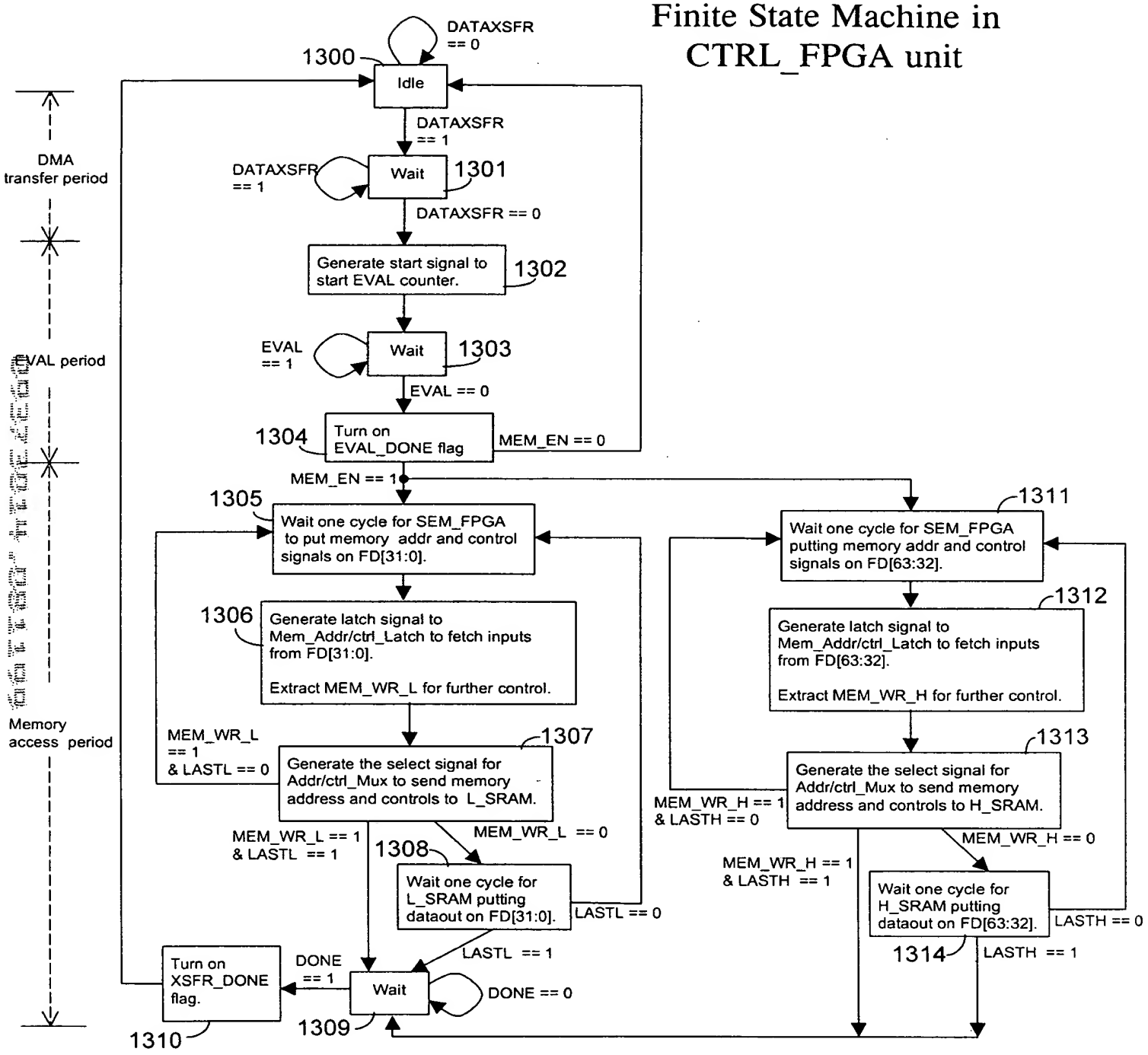


FIG. 58

EVALFSM - EVAL
Finite State Machine in each FPGA logic device

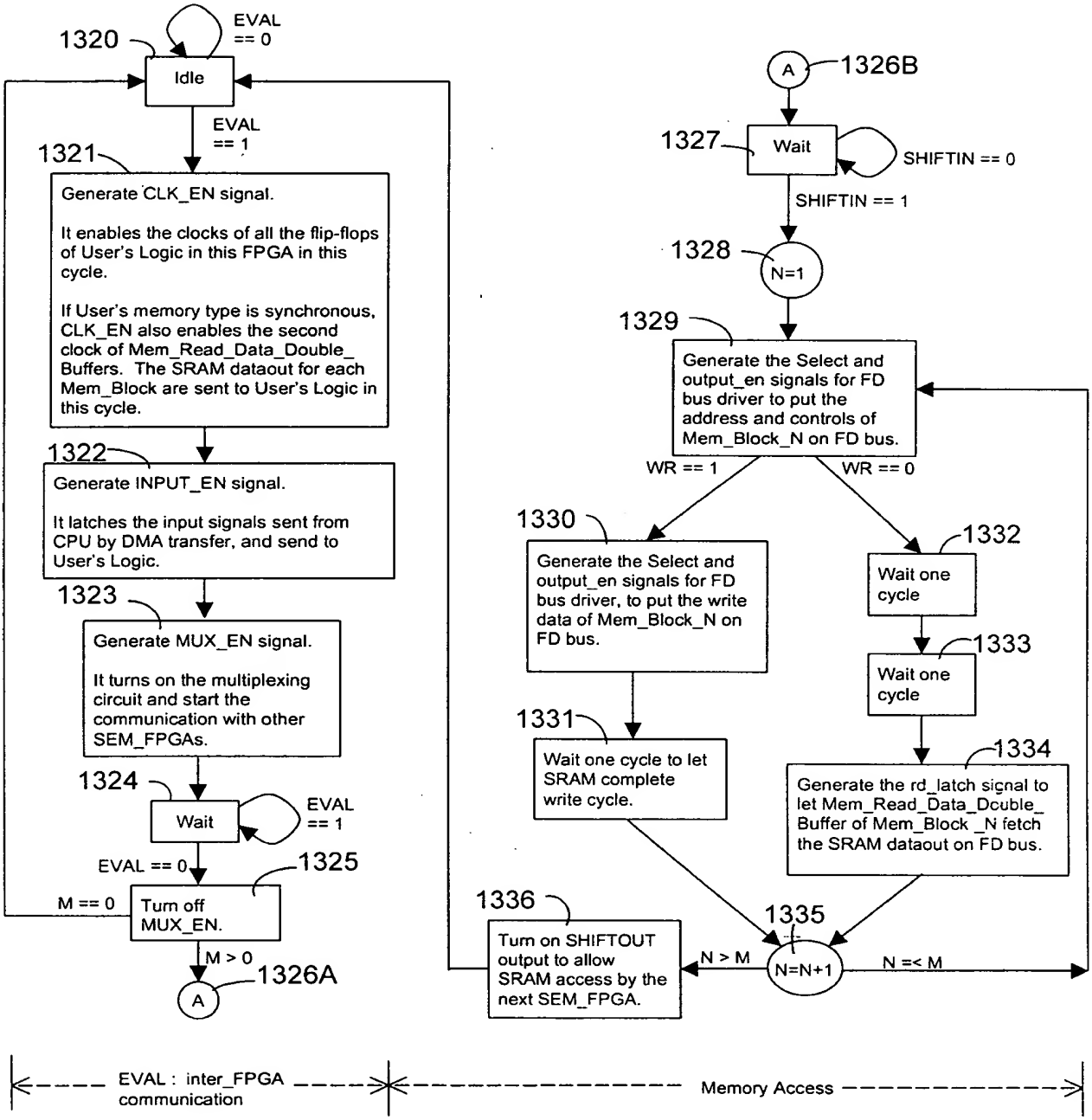


FIG. 59

MEMORY READ DATA DOUBLE BUFFER

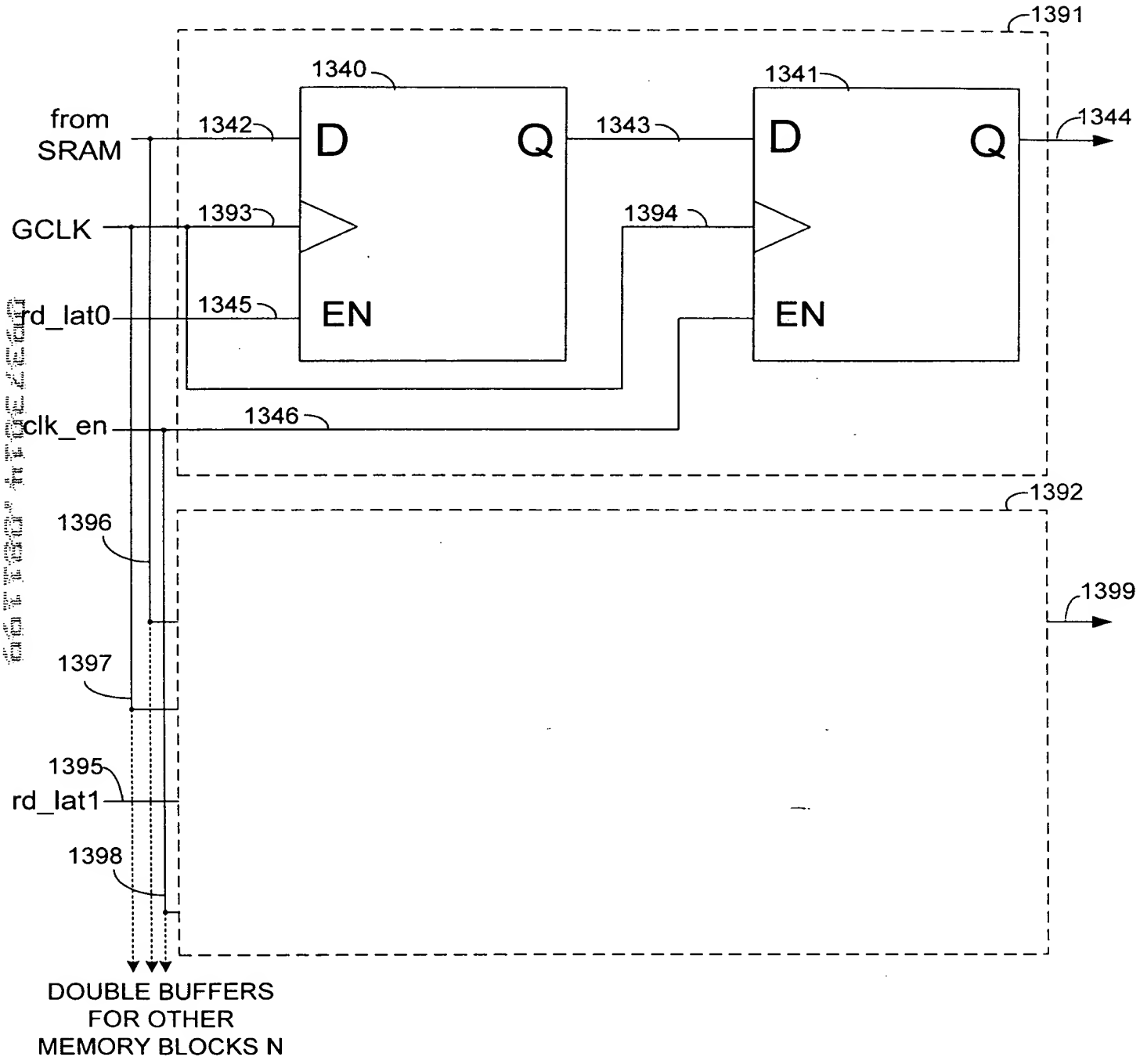


FIG. 60

SIMULATION WRITE/READ CYCLE

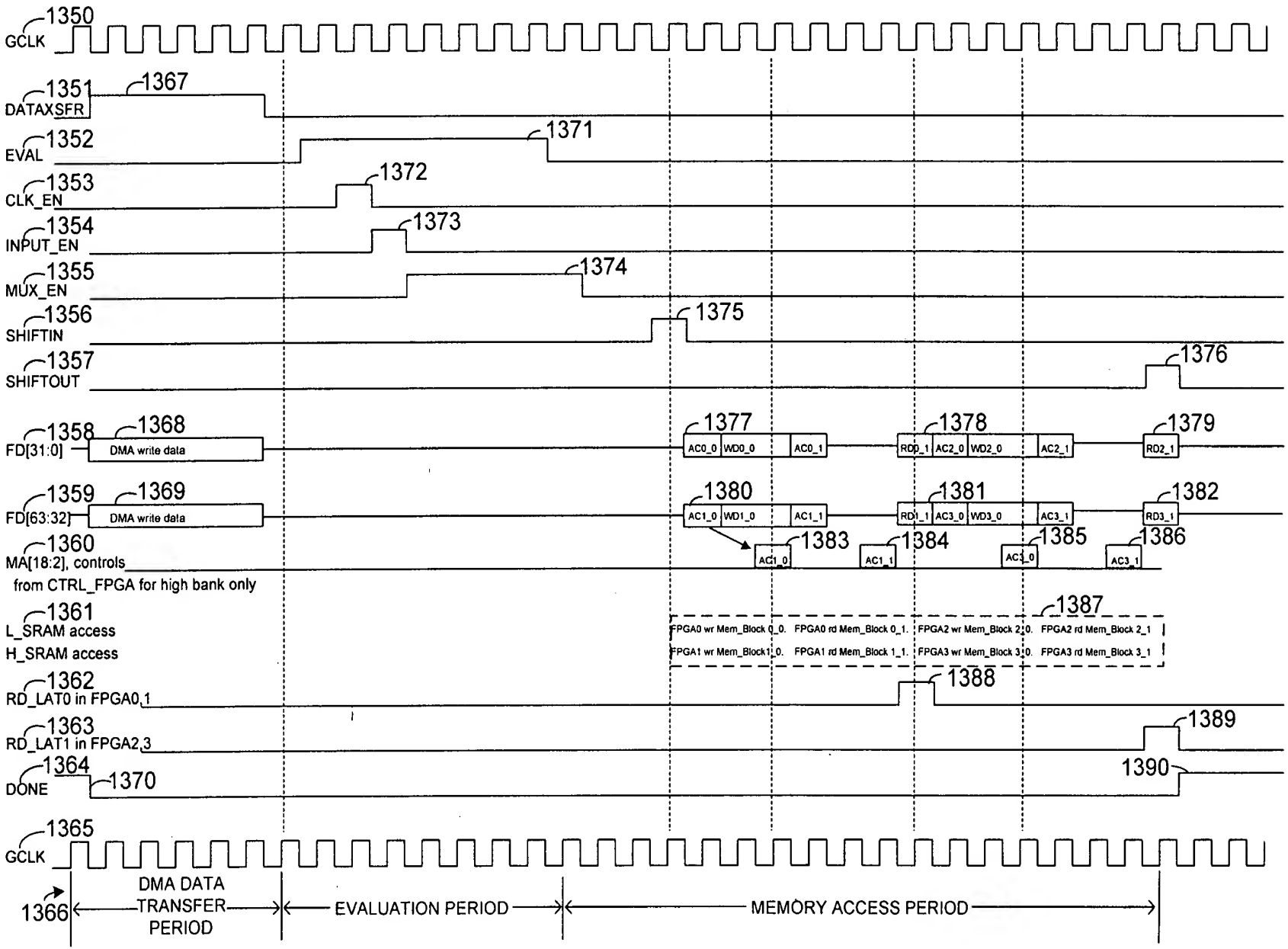


FIG. 61

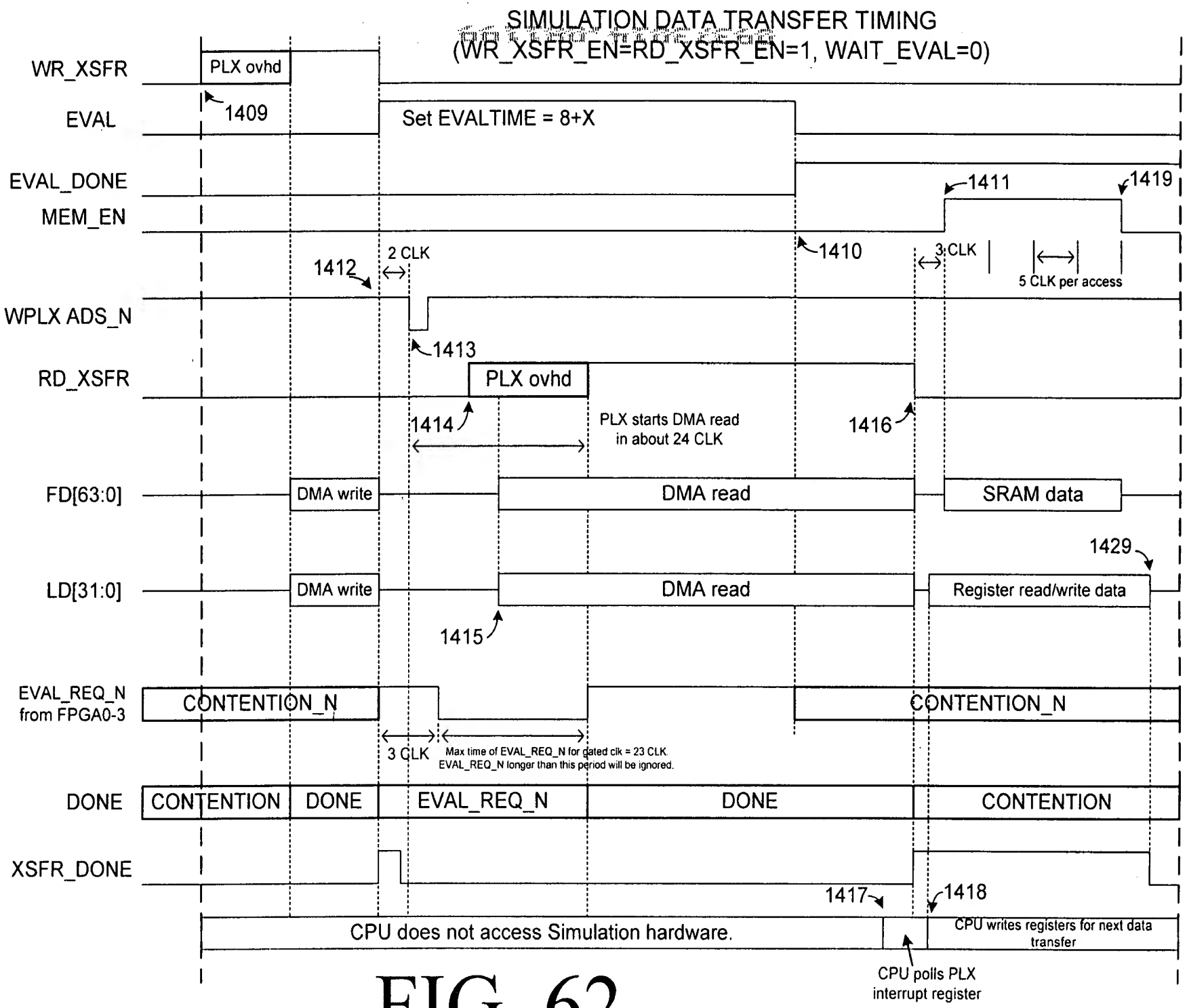

SIMULATION DATA TRANSFER TIMING
(WR_XSFR_EN=RD_XSFR_EN=1, WAIT_EVAL=0)
WR_XSFR
PLX ovhd
EVAL
1409
Set EVALTIME = 8+X
EVAL_DONE
1411
1419
MEM_EN
1410
3 CLK
5 CLK per access
1412
2 CLK
WPLX ADS_N
1413
RD_XSFR
PLX ovhd
1414
PLX starts DMA read
in about 24 CLK
1416
FD[63:0]
DMA write
DMA read
SRAM data
1429
LD[31:0]
DMA write
DMA read
Register read/write data
1415
EVAL_REQ_N
from FPGA0-3
CONTENTION_N
CONTENTION_N
3 CLK
Max time of EVAL_REQ_N for gated clk = 23 CLK
EVAL_REQ_N longer than this period will be ignored.
DONE
CONTENTION
DONE
EVAL_REQ_N
DONE
CONTENTION
XSFR_DONE
1417
1418
CPU does not access Simulation hardware.
CPU writes registers for next data transfer
CPU polls PLX
interrupt register

FIG. 62

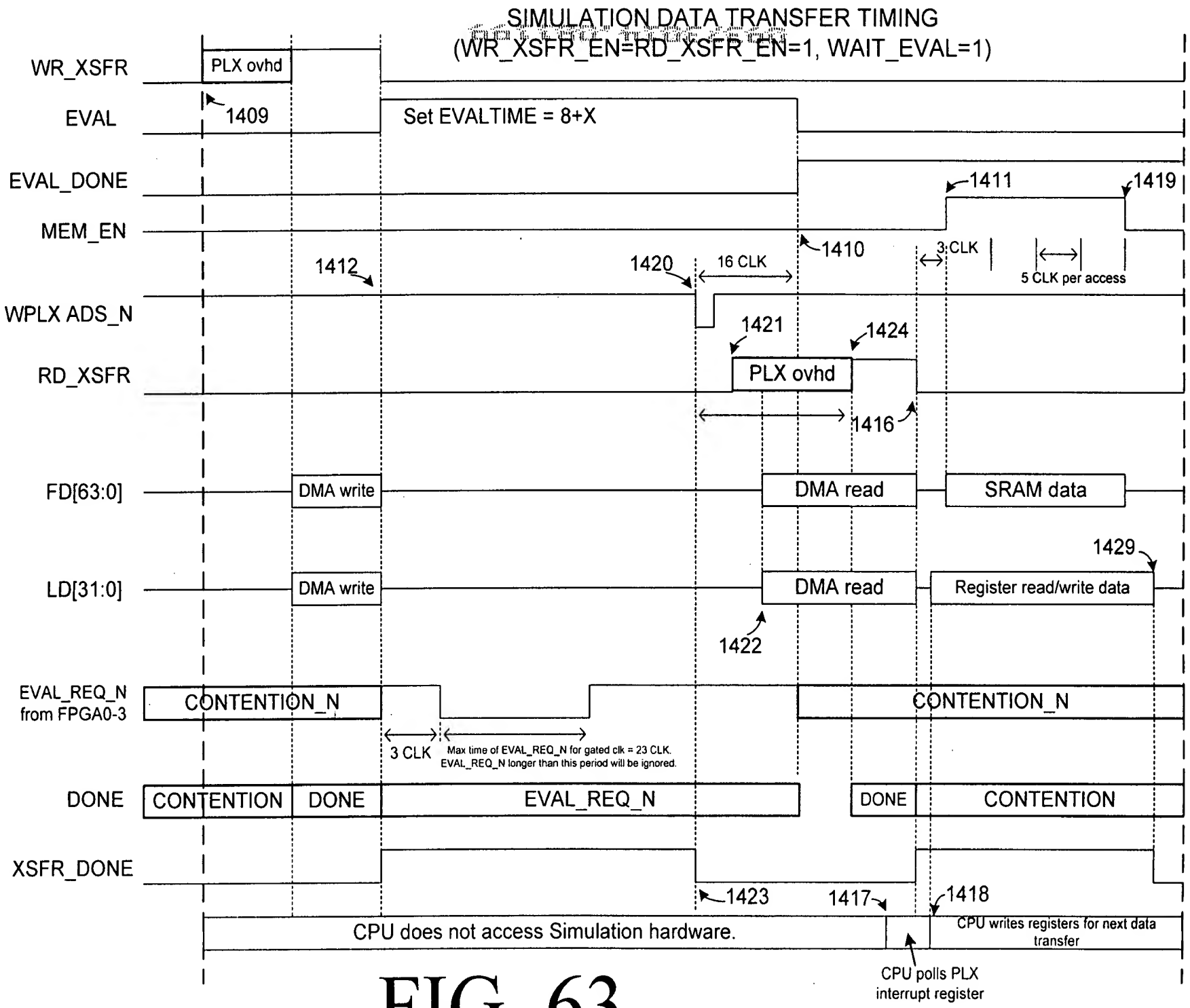

SIMULATION DATA TRANSFER TIMING
(WR_XSFR_EN=RD_XSFR_EN=1, WAIT_EVAL=1)
WR_XSFR
PLX ovhd
EVAL
1409
Set EVALTIME = 8+X
EVAL_DONE
1411
1419
MEM_EN
1410
3 CLK
5 CLK per access
1412
1420
16 CLK
WPLX ADS_N
1421
1424
RD_XSFR
PLX ovhd
1416
FD[63:0]
DMA write
DMA read
SRAM data
1429
LD[31:0]
DMA write
DMA read
Register read/write data
1422
EVAL_REQ_N
from FPGA0-3
CONTENTION_N
CONTENTION_N
3 CLK
Max time of EVAL_REQ_N for gated clk = 23 CLK.
EVAL_REQ_N longer than this period will be ignored.
DONE
CONTENTION
DONE
EVAL_REQ_N
DONE
CONTENTION
XSFR_DONE
1423
1417
1418
CPU does not access Simulation hardware.
CPU writes registers for next data transfer
CPU polls PLX
interrupt register

FIG. 63

Typical User Design of PCI Add-on Cards

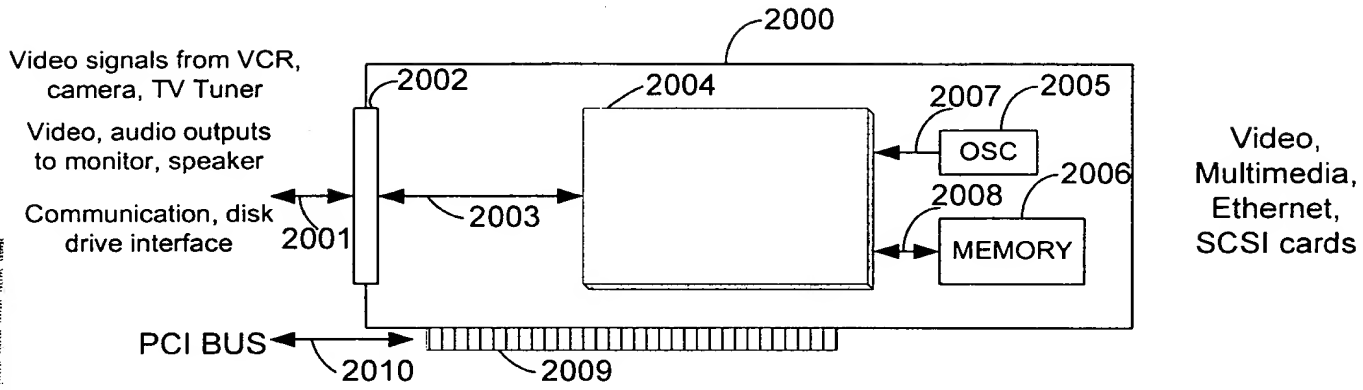


FIG. 64

Typical Hardware/Software Co-Verification

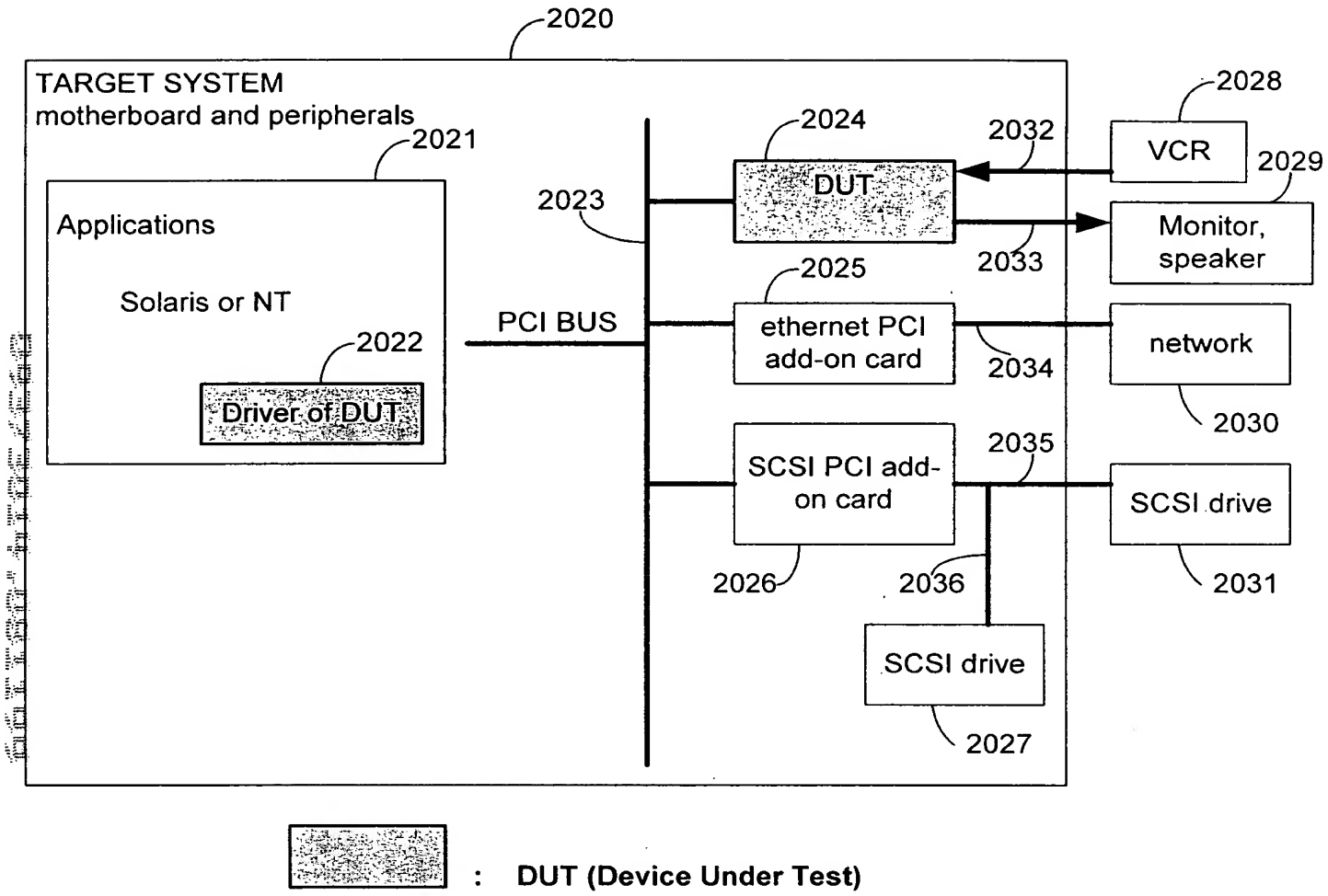
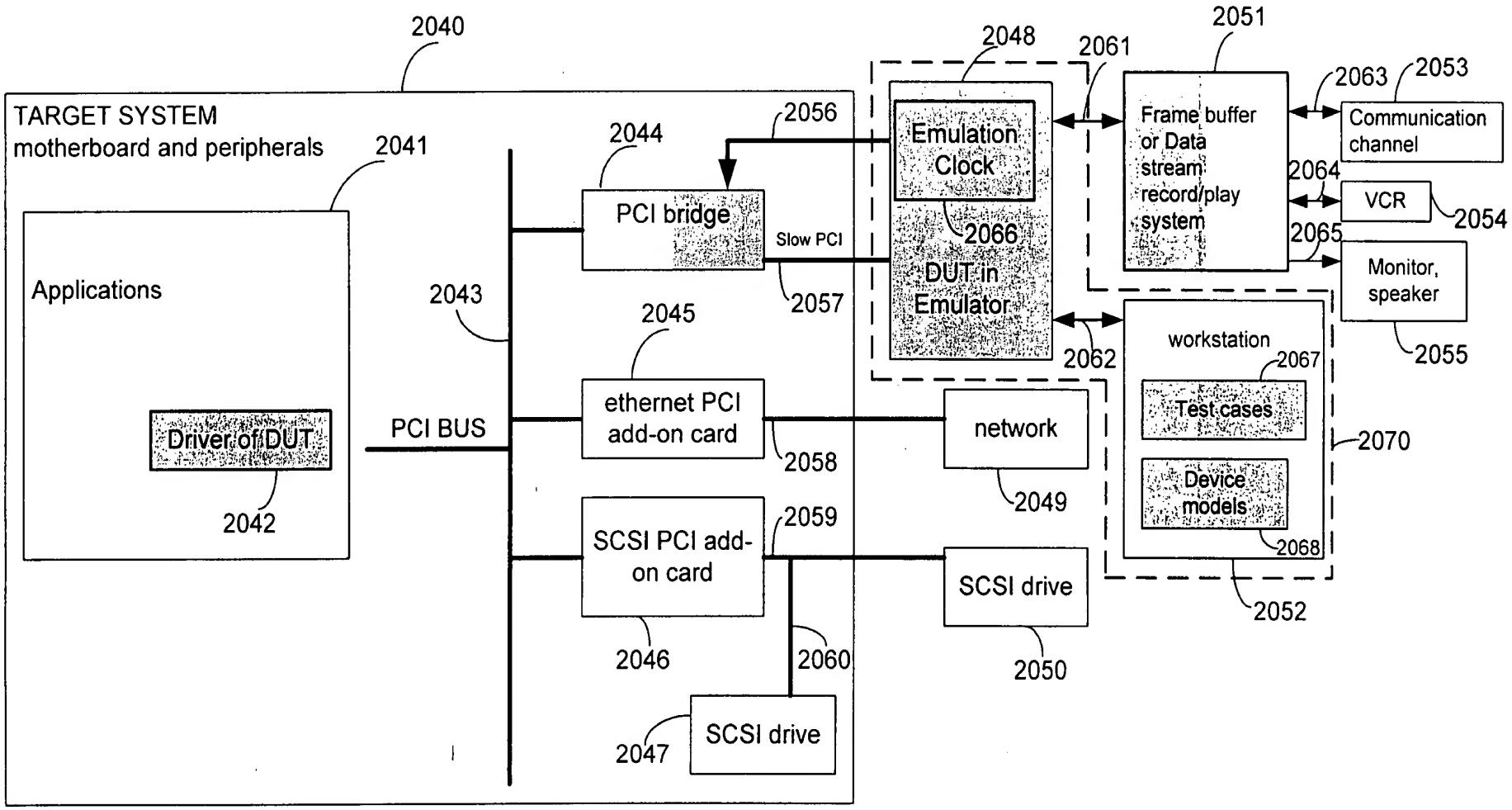


FIG. 65

Typical Co-Verification by Using Emulator



: **running time at emulation speed**

The rest of the target system is running at full speed.

FIG. 66

SIMULATION

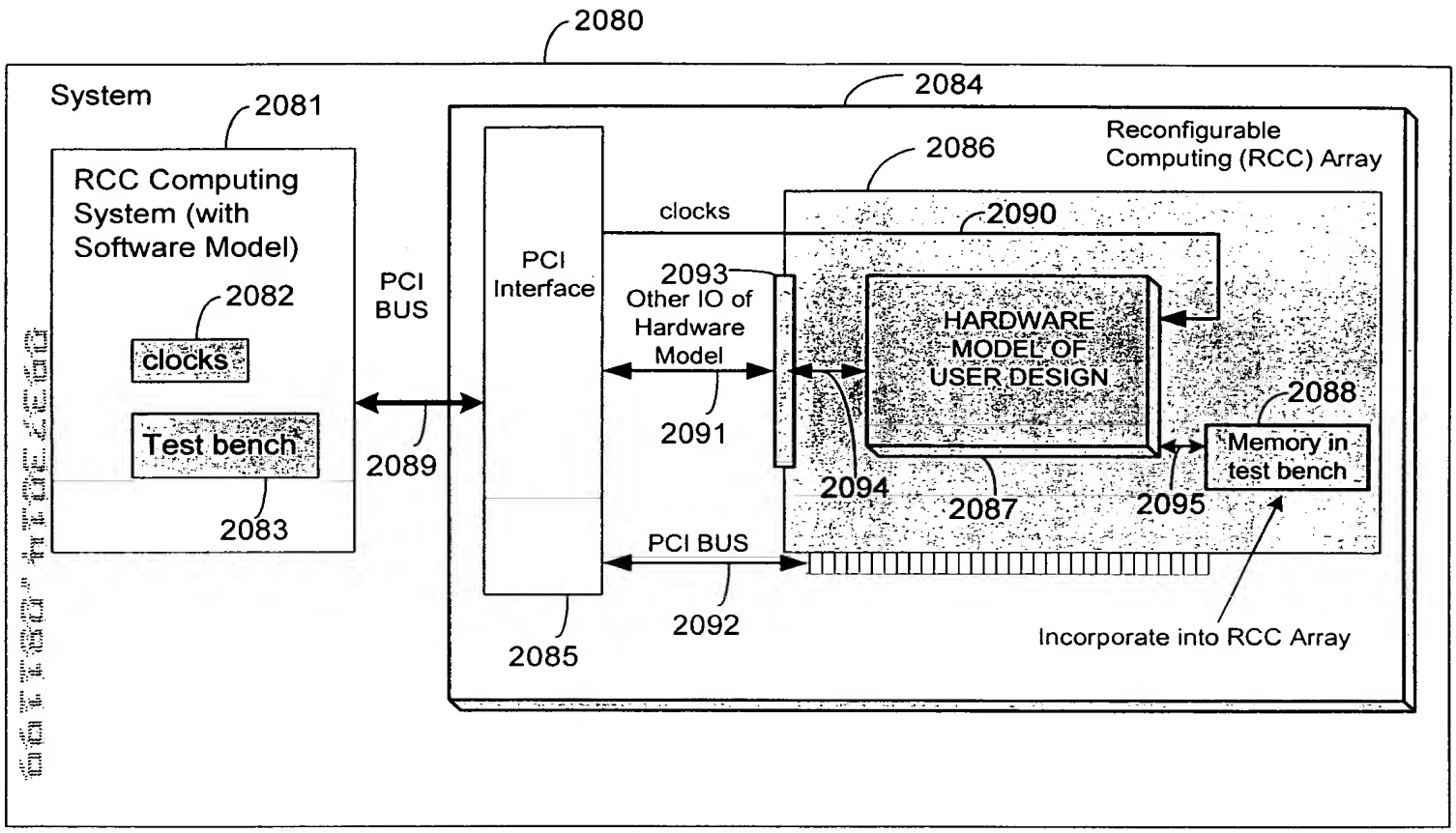


FIG. 67

CO-VERIFICATION WITHOUT EXTERNAL I/O

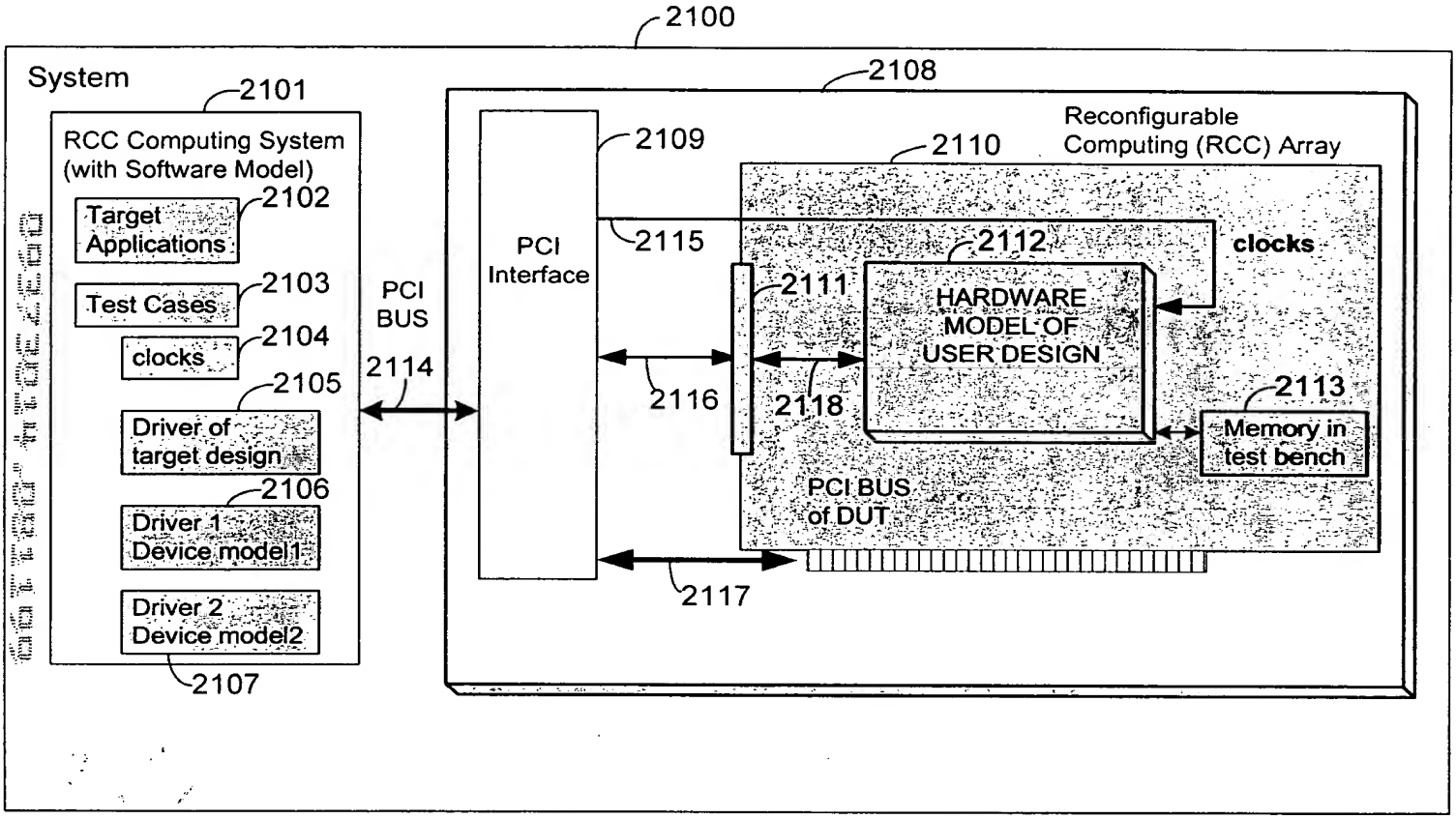


FIG. 68

CO-VERIFICATION WITH EXTERNAL I/O

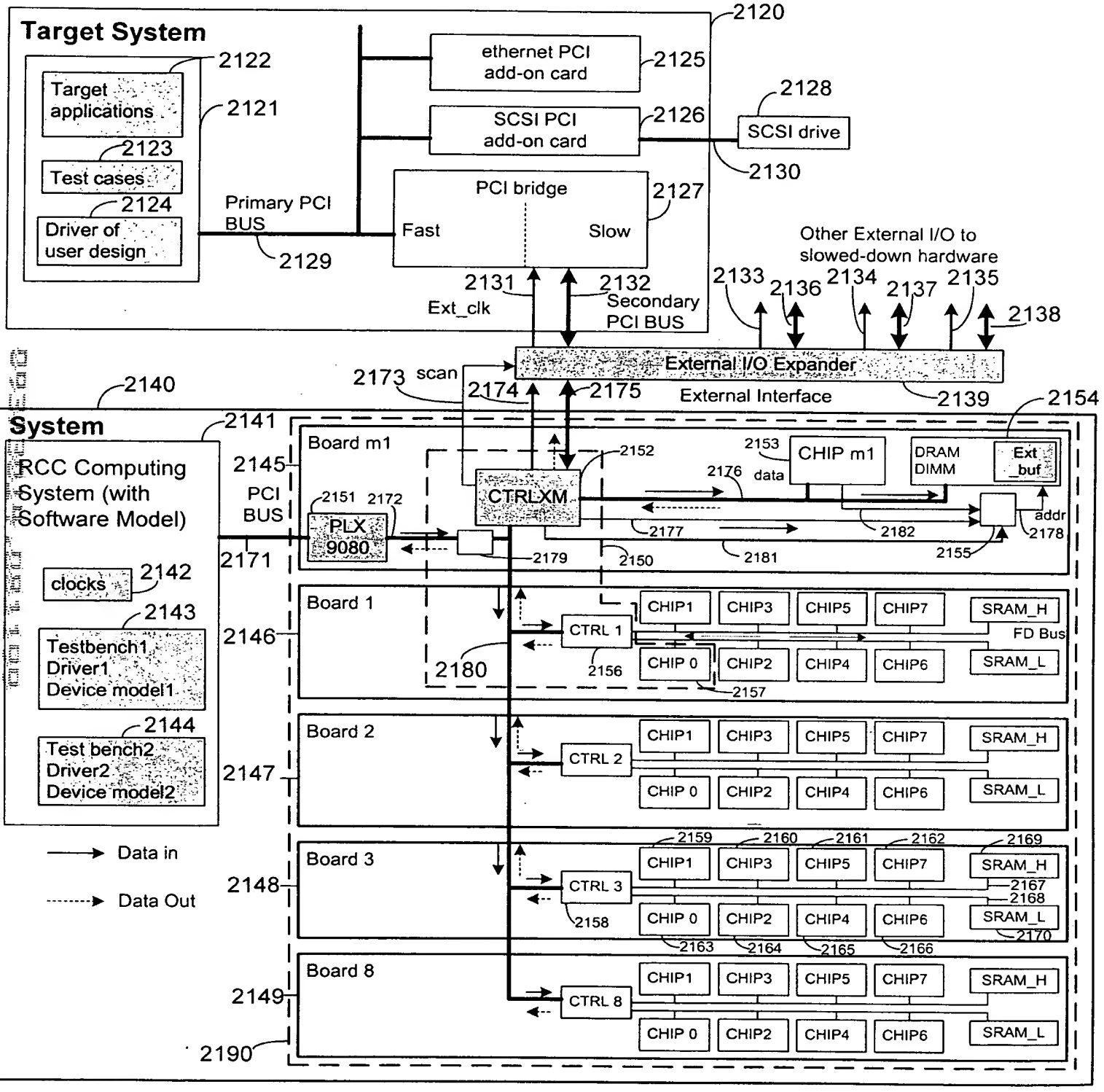


FIG. 69

CONTROL OF DATA-IN CYCLE

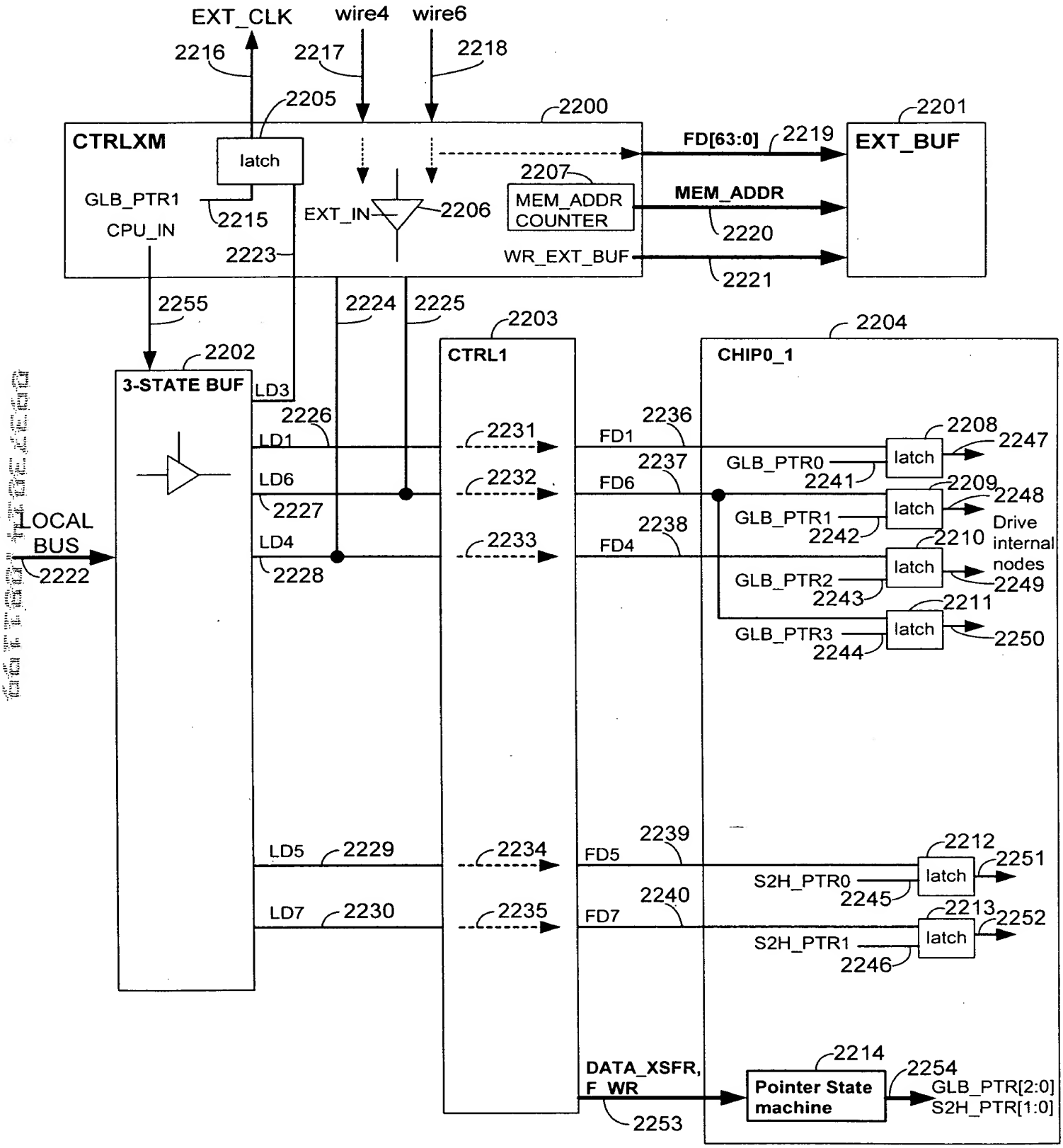


FIG. 70

CONTROL OF DATA-OUT CYCLE

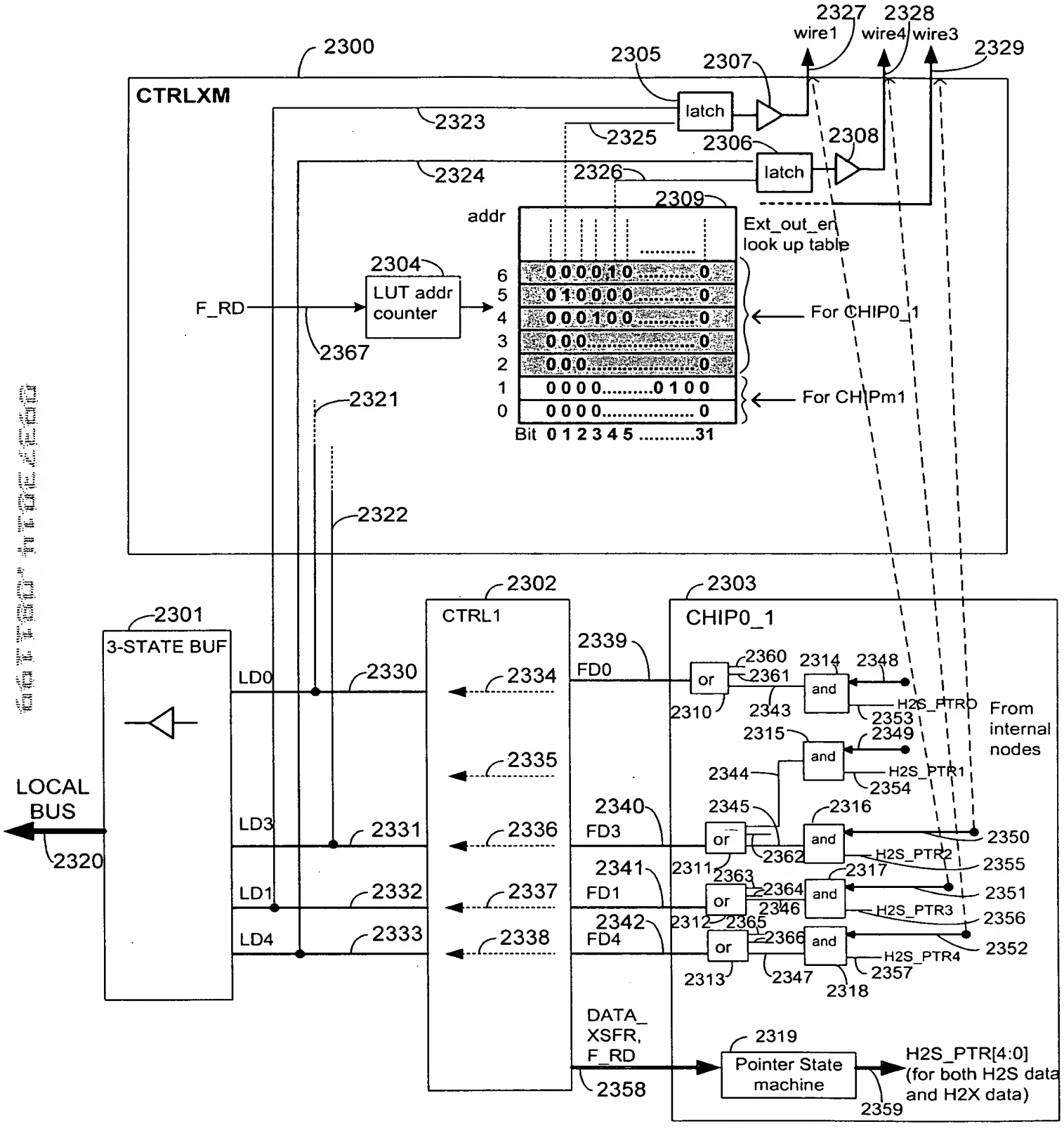


FIG. 71

CONTROL OF DATA-IN CYCLE

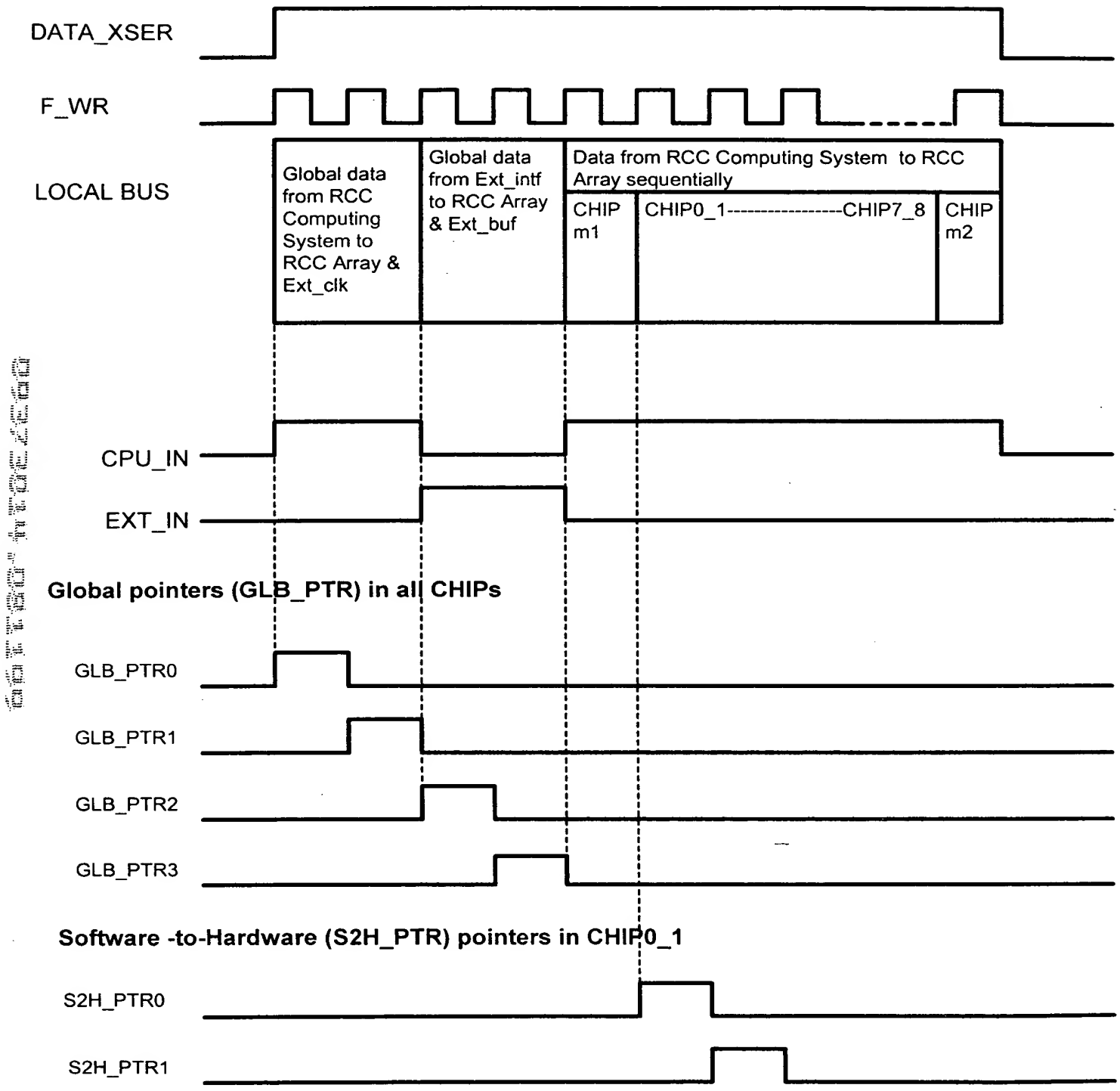


FIG. 72

CONTROL OF DATA-OUT CYCLE

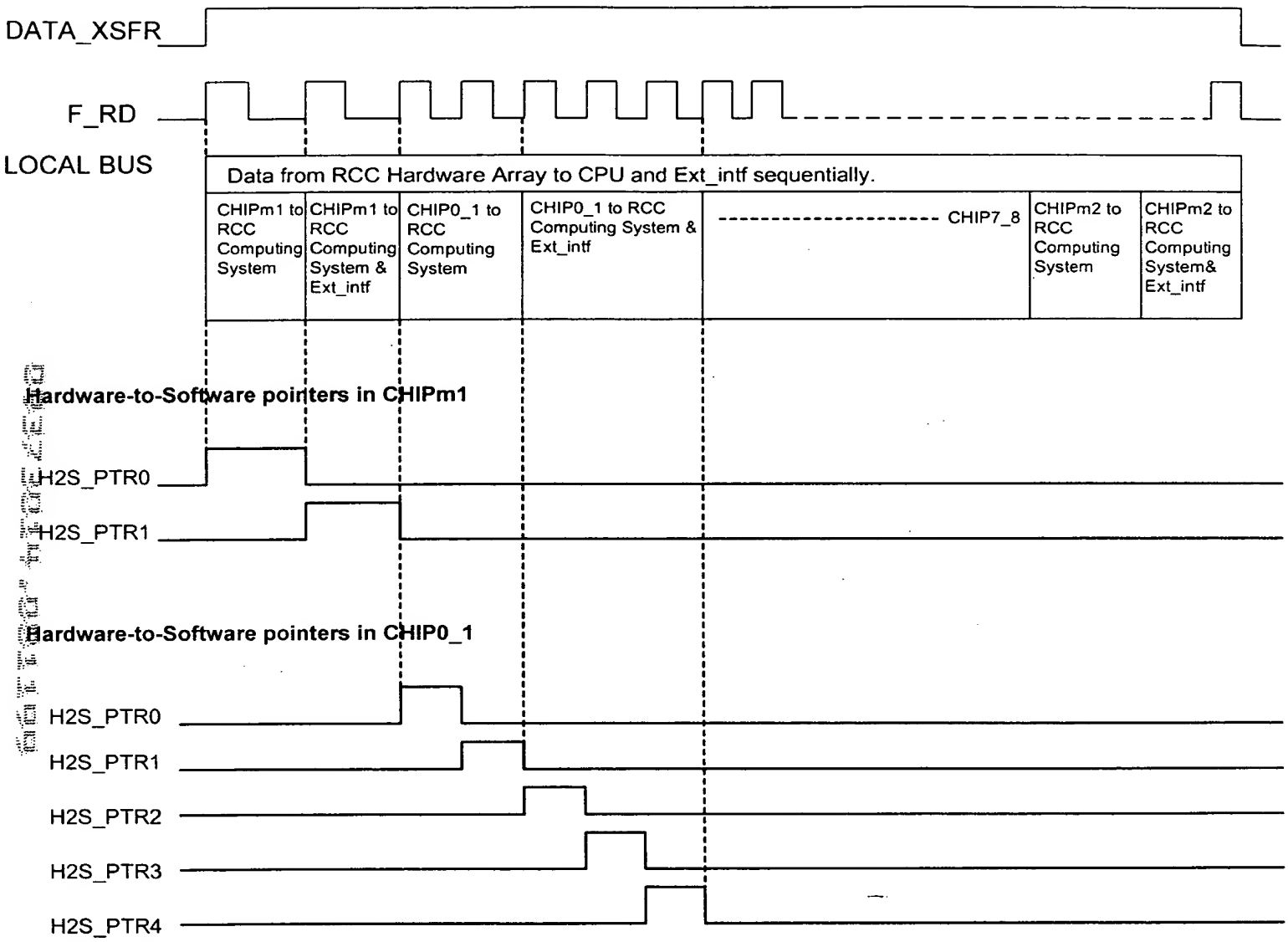


FIG. 73

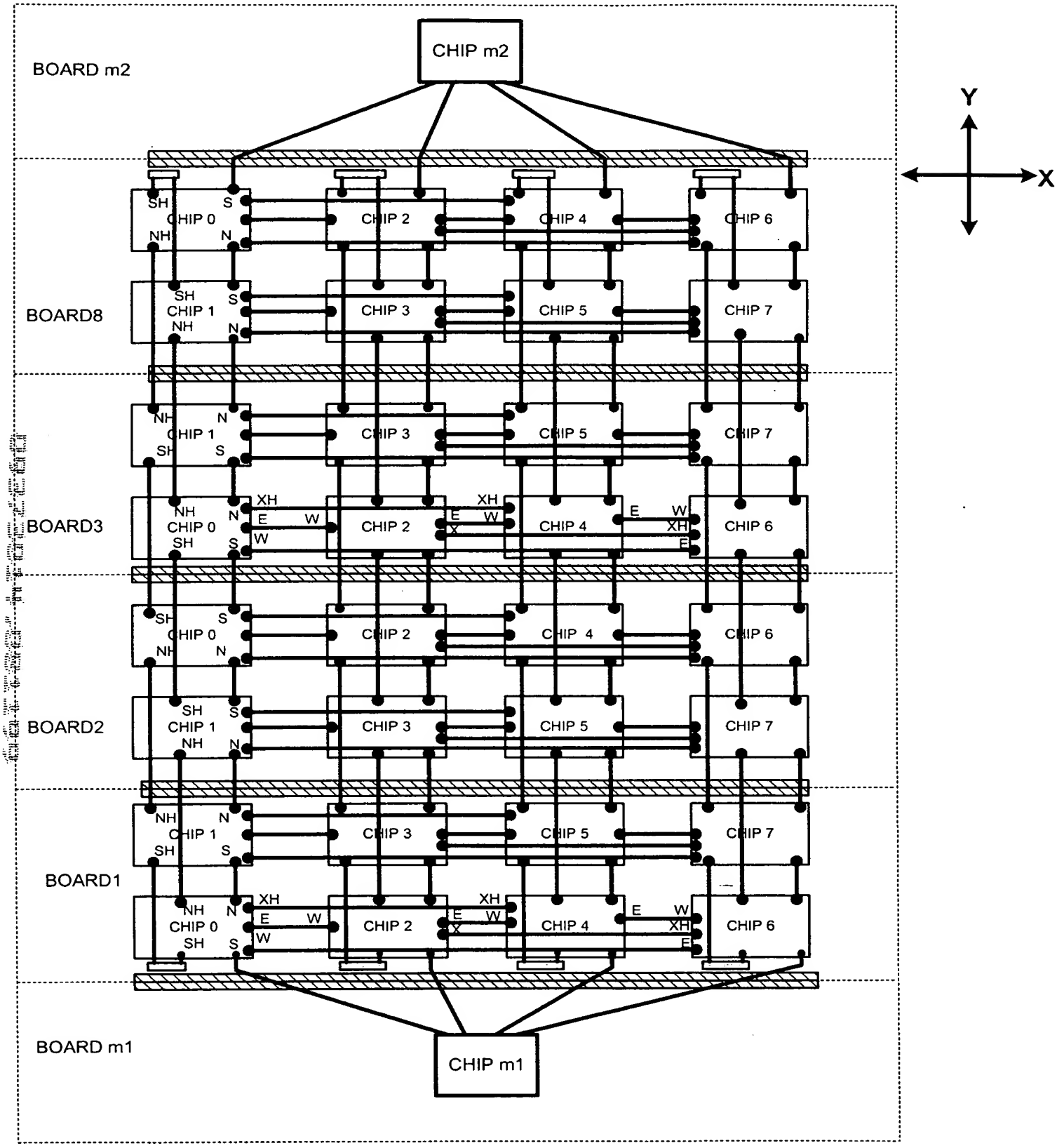

BOARD m2
CHIP m2
Y
X
BOARD8
BOARD3
BOARD2
BOARD1
BOARD m1
CHIP m1
CHIP 0
CHIP 1
CHIP 2
CHIP 3
CHIP 4
CHIP 5
CHIP 6
CHIP 7
SH
NH
S
N
XH
E
W
X

FIG. 74

SHIFT REGISTER

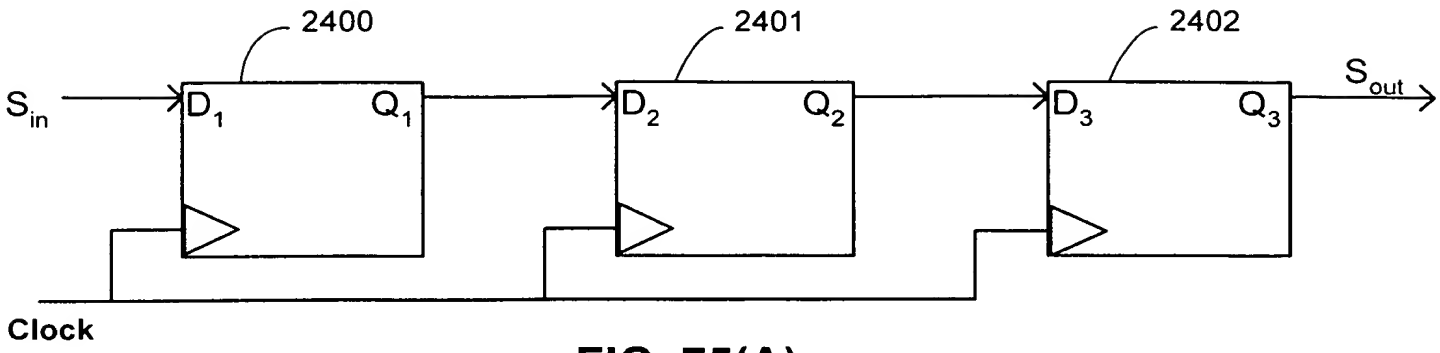


FIG. 75(A)

HOLD TIME ASSUMPTION FOR SHIFT REGISTER

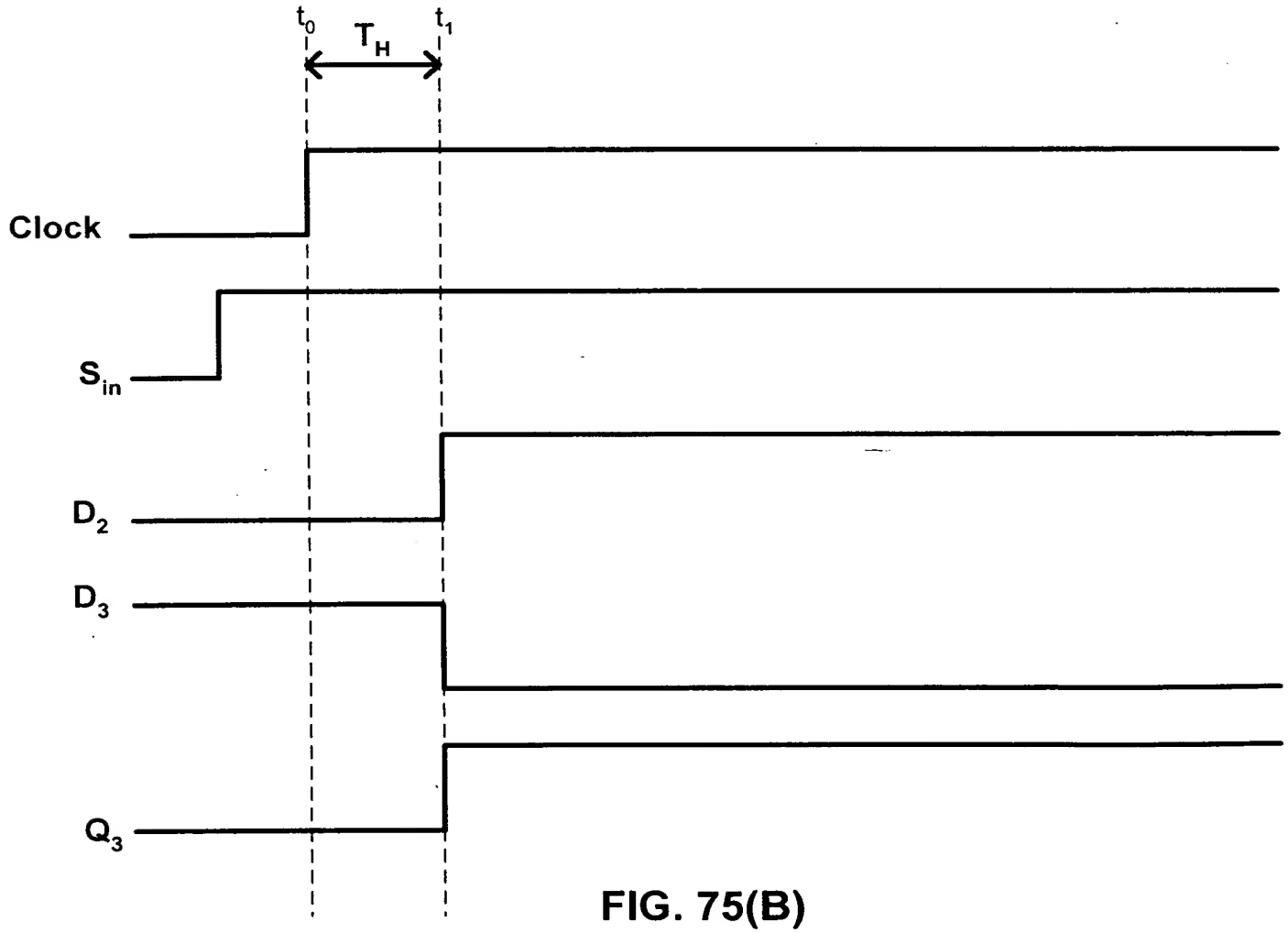


FIG. 75(B)

MULTIPLE FPGA MAPPING FOR SHIFT REGISTER

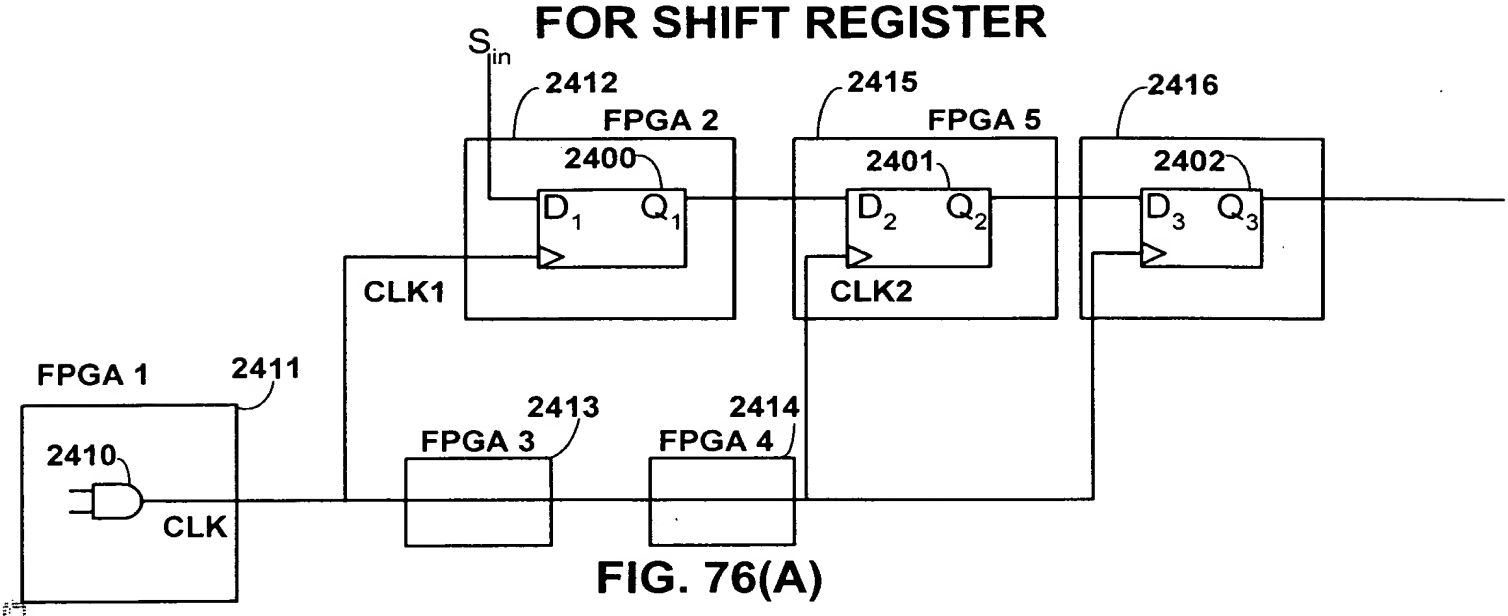


FIG. 76(A)

HOLD TIME VIOLATION BY LONG CLOCK SKEW

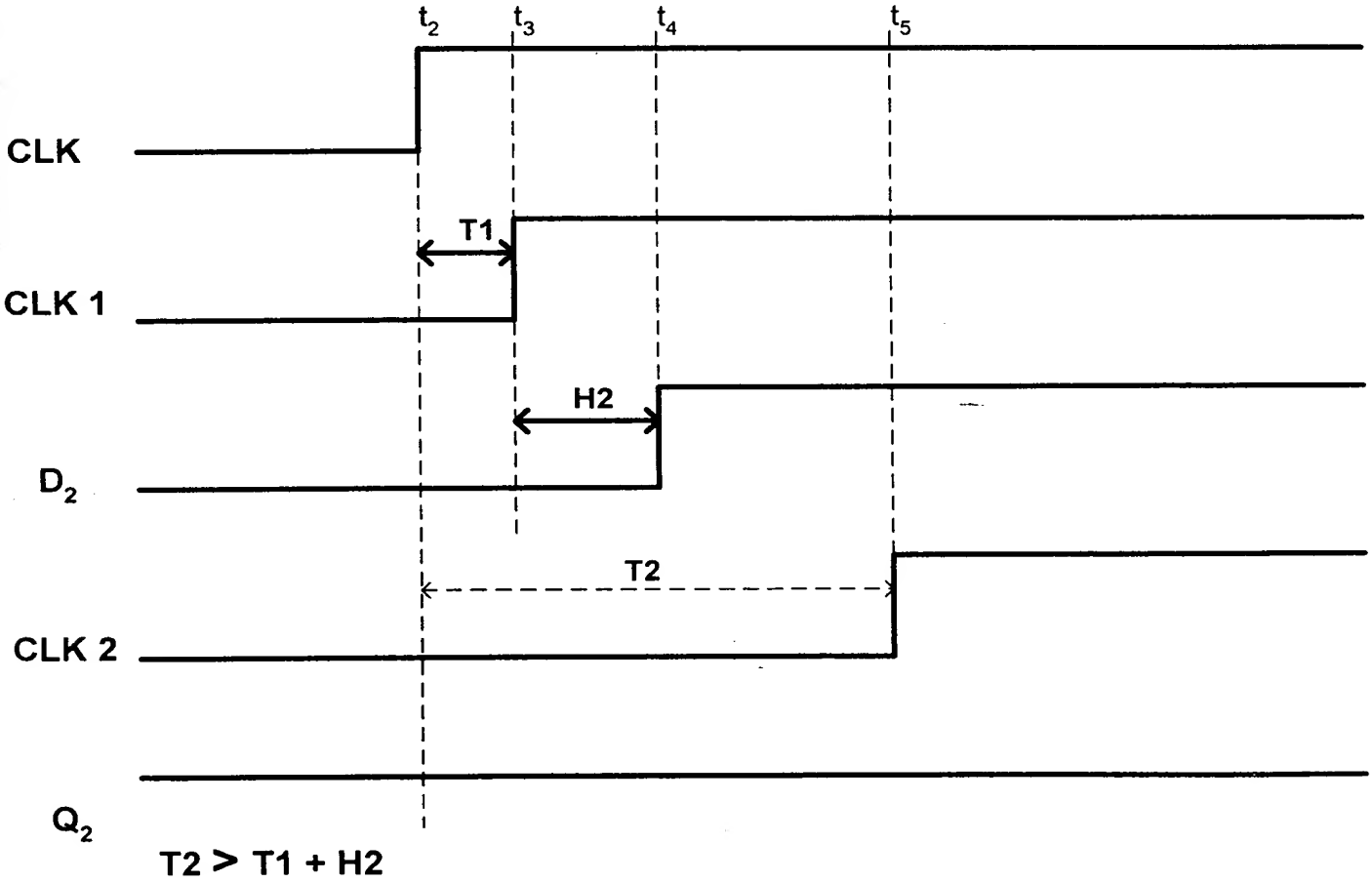


FIG. 76(B)

CLOCK GLITCH PROBLEM

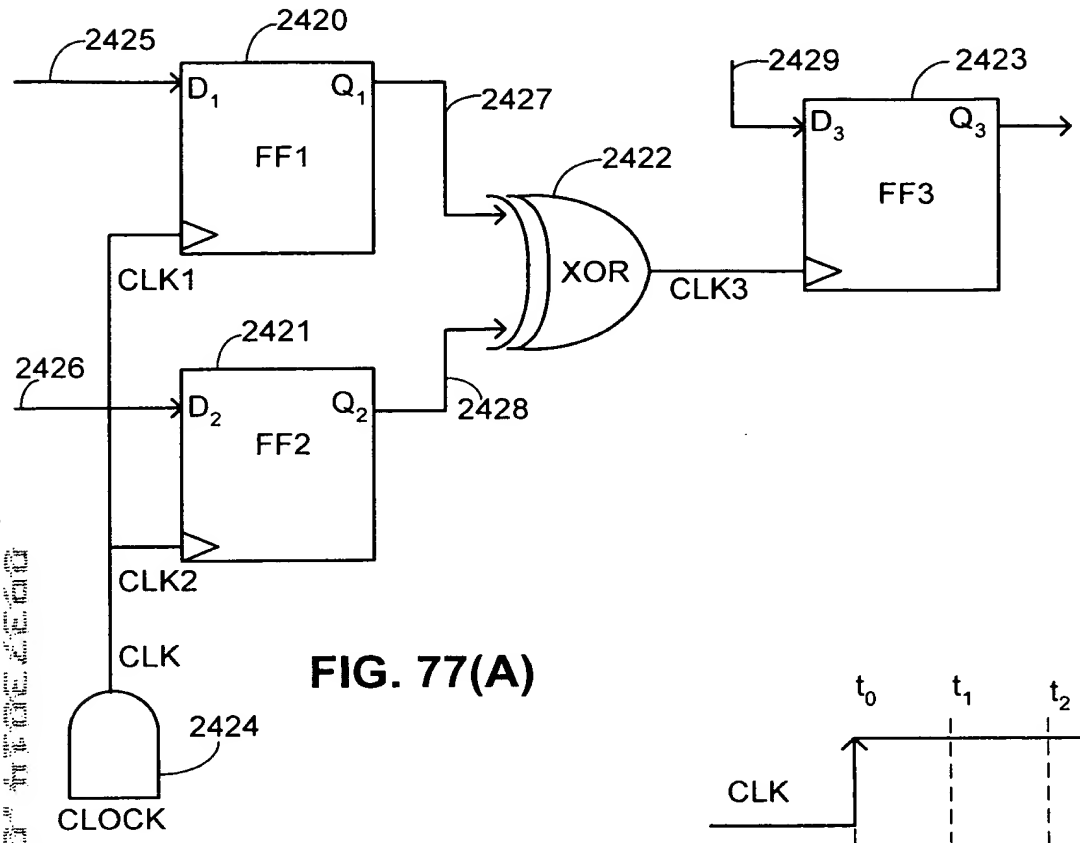


FIG. 77(A)

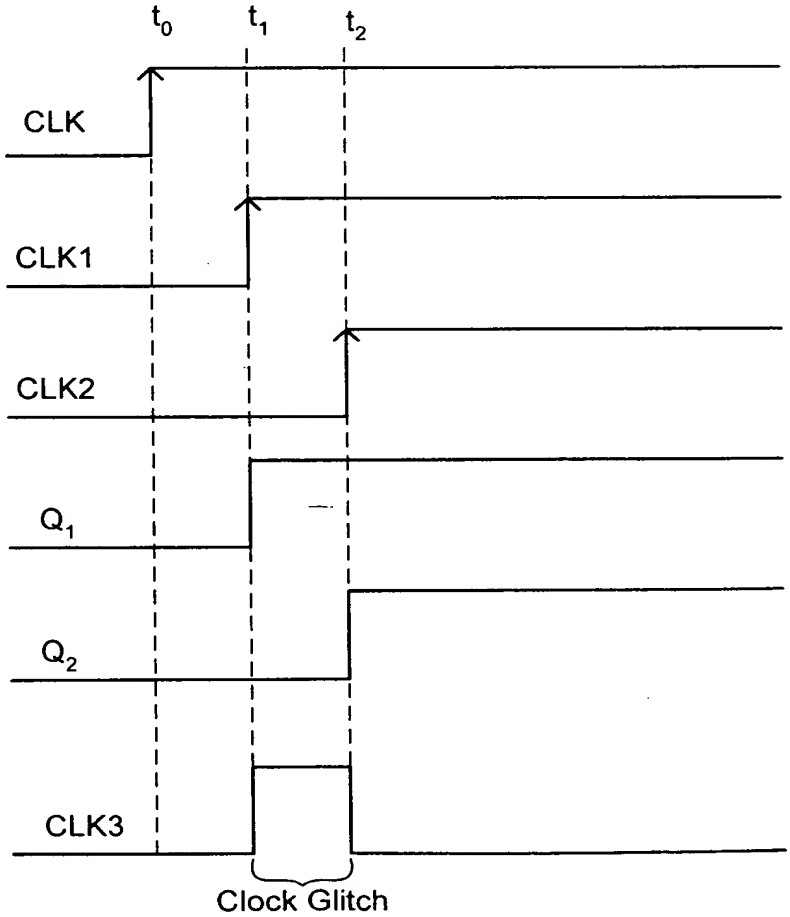
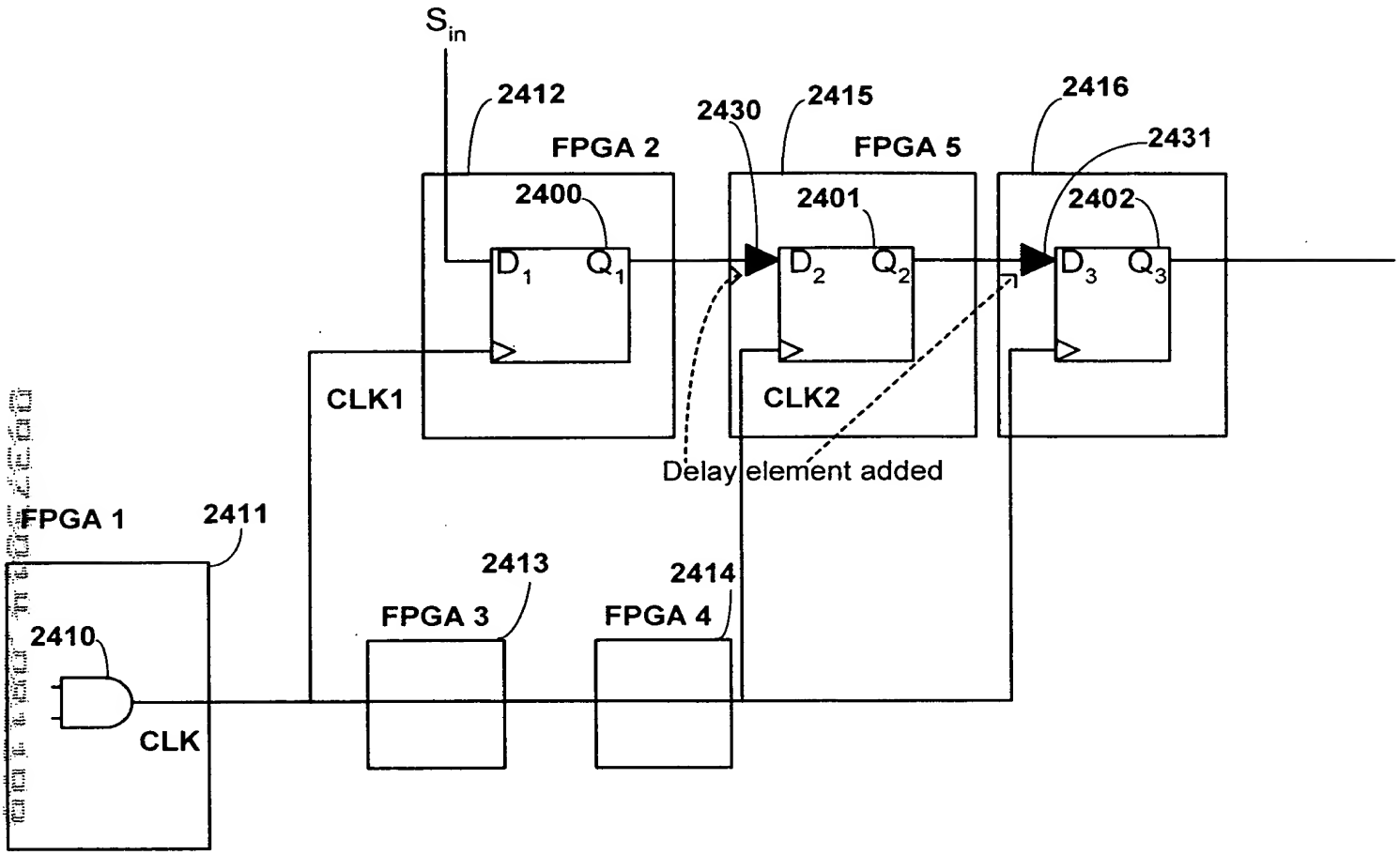


FIG. 77(B)

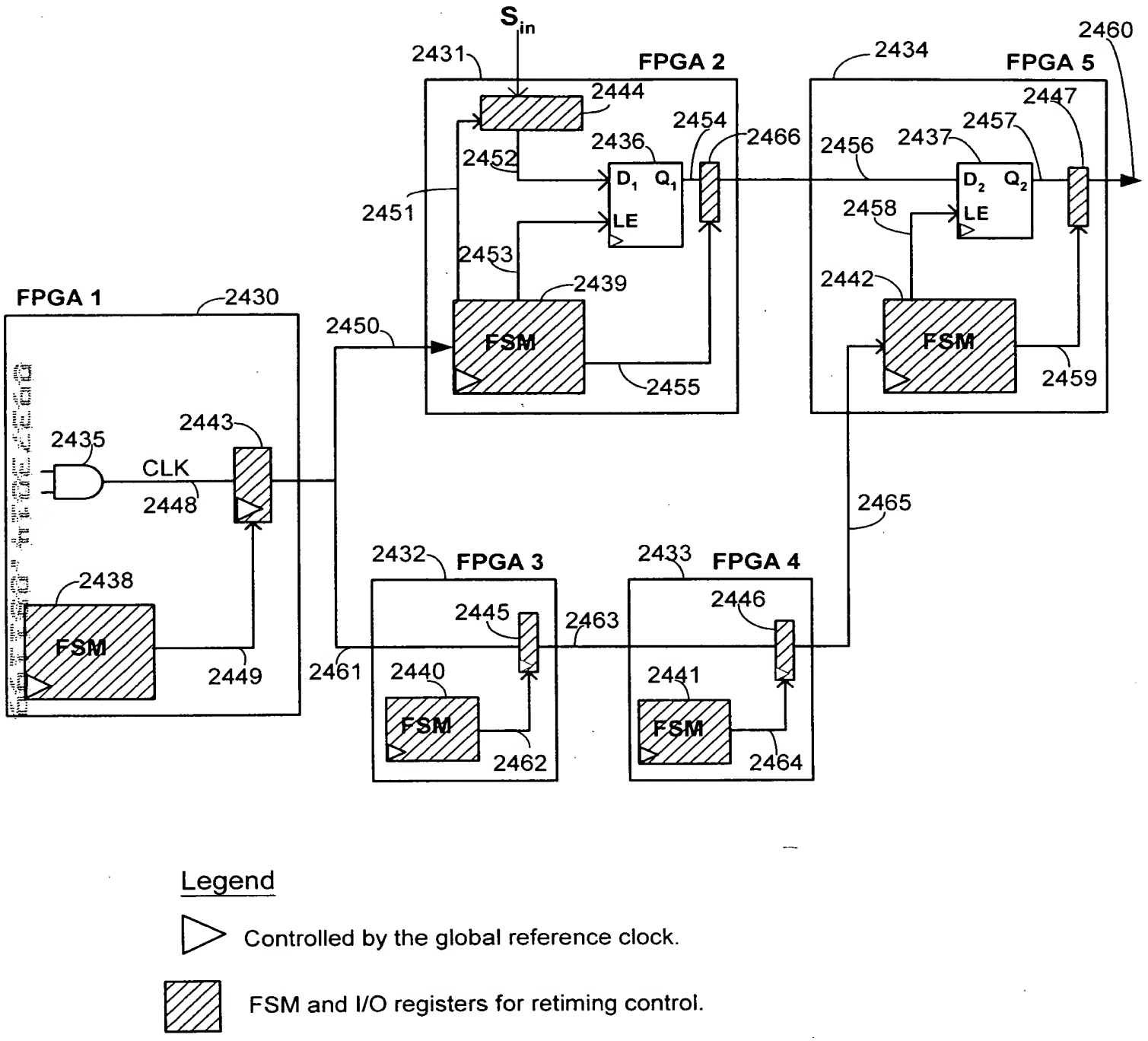
TIMING ADJUSTMENT BY ADDING DELAY



(Prior Art)

FIG. 78

GLOBAL RETIMING



Legend

Controlled by the global reference clock.

FSM and I/O registers for retiming control.

(Prior Art)

FIG. 79

TIGF LATCH

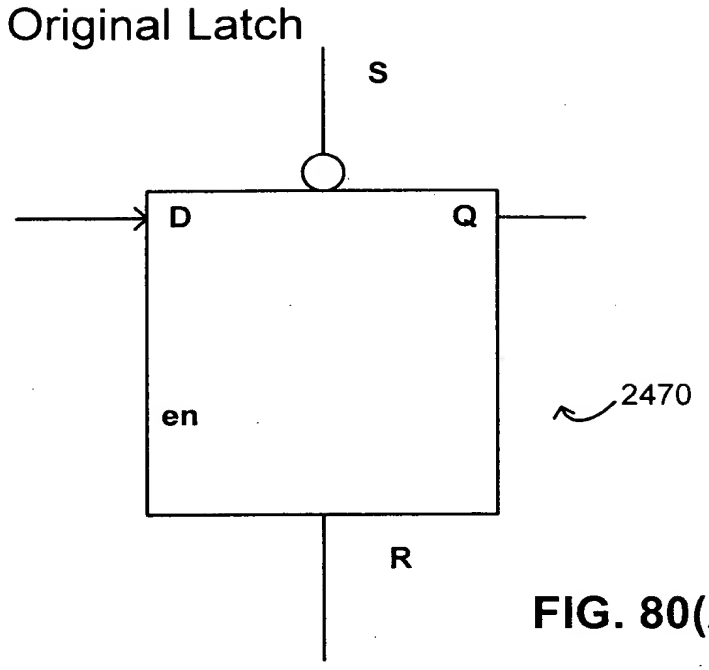


FIG. 80(A)

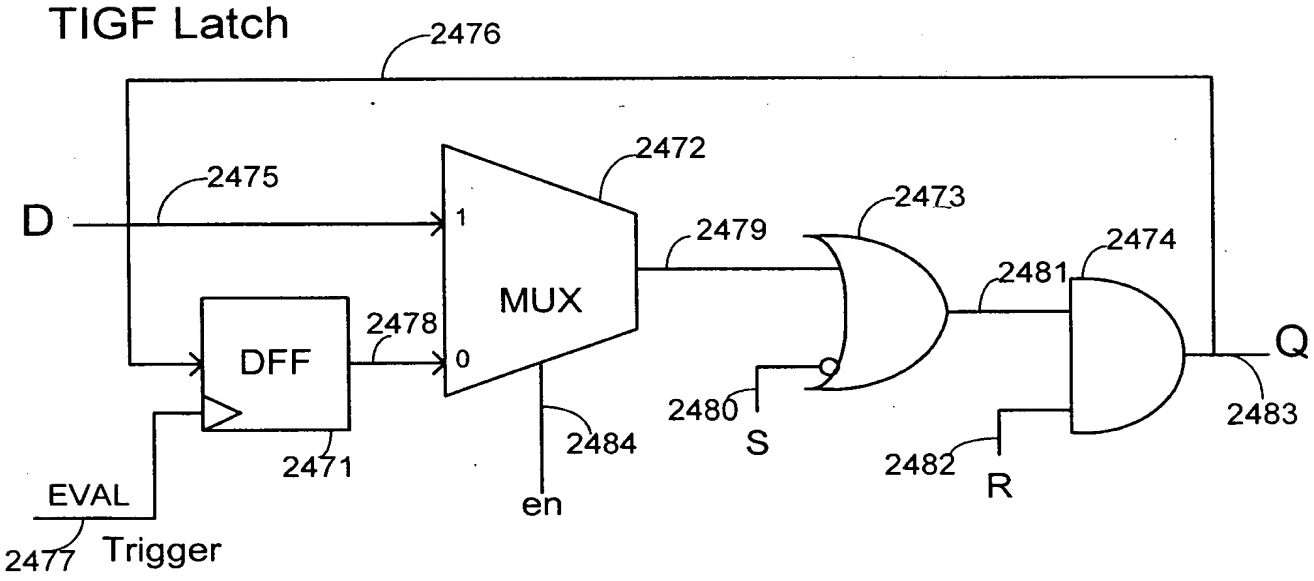


FIG. 80(B)

TIGF DFF

Original DFF

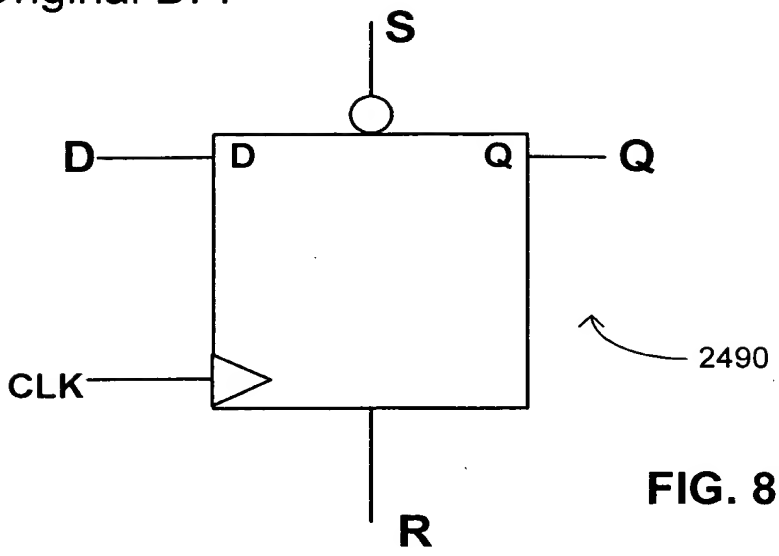


FIG. 81(A)

TIGF DFF and Edge Detector

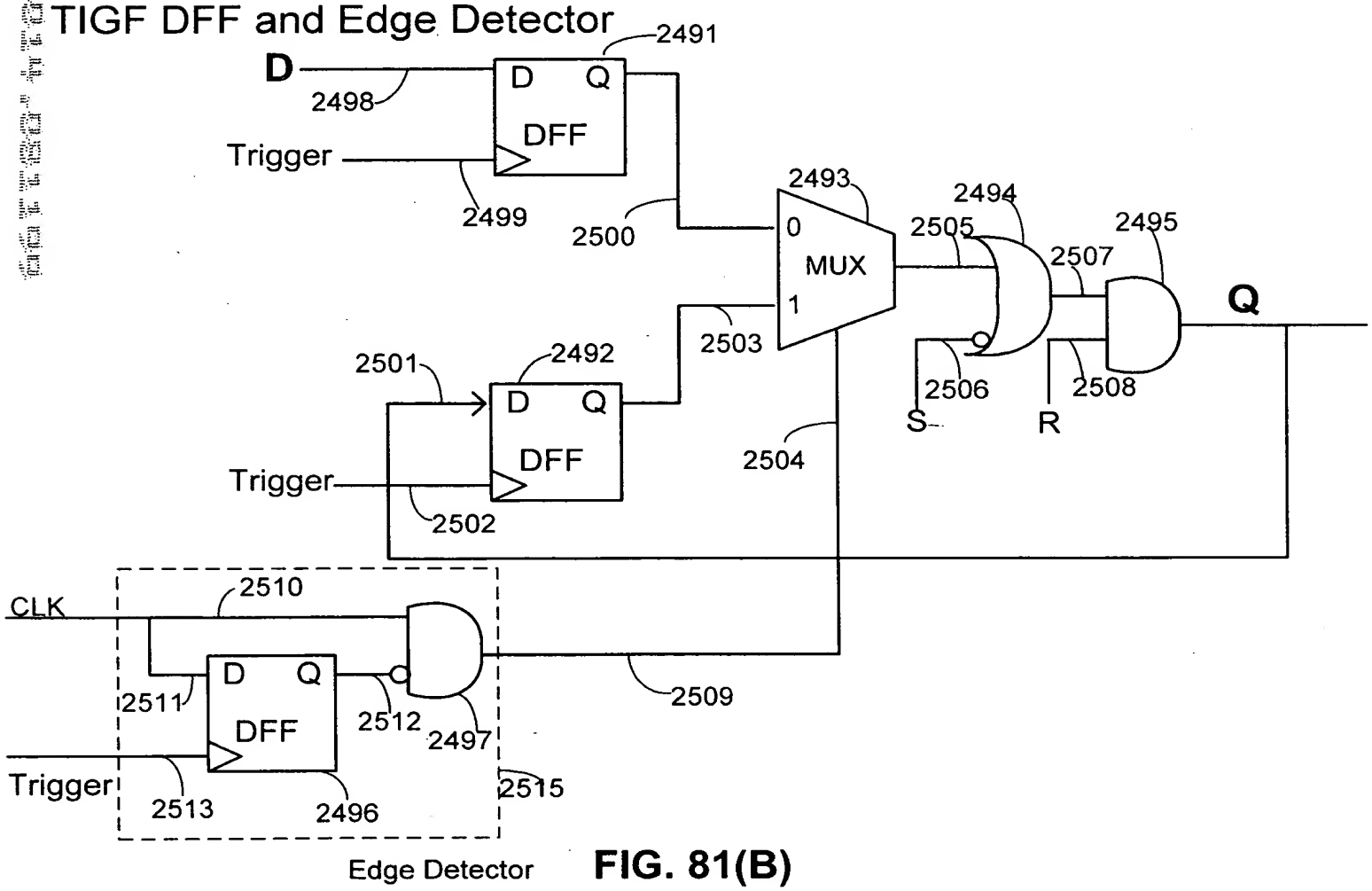


FIG. 81(B)

GLOBAL TRIGGER SIGNAL

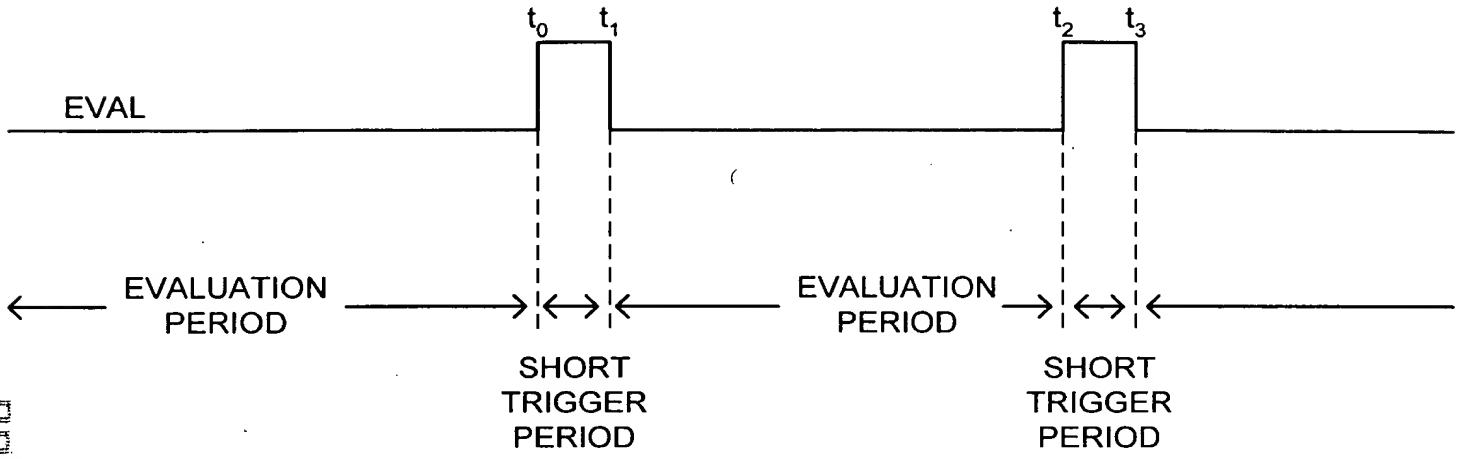


FIG. 82

RCC System

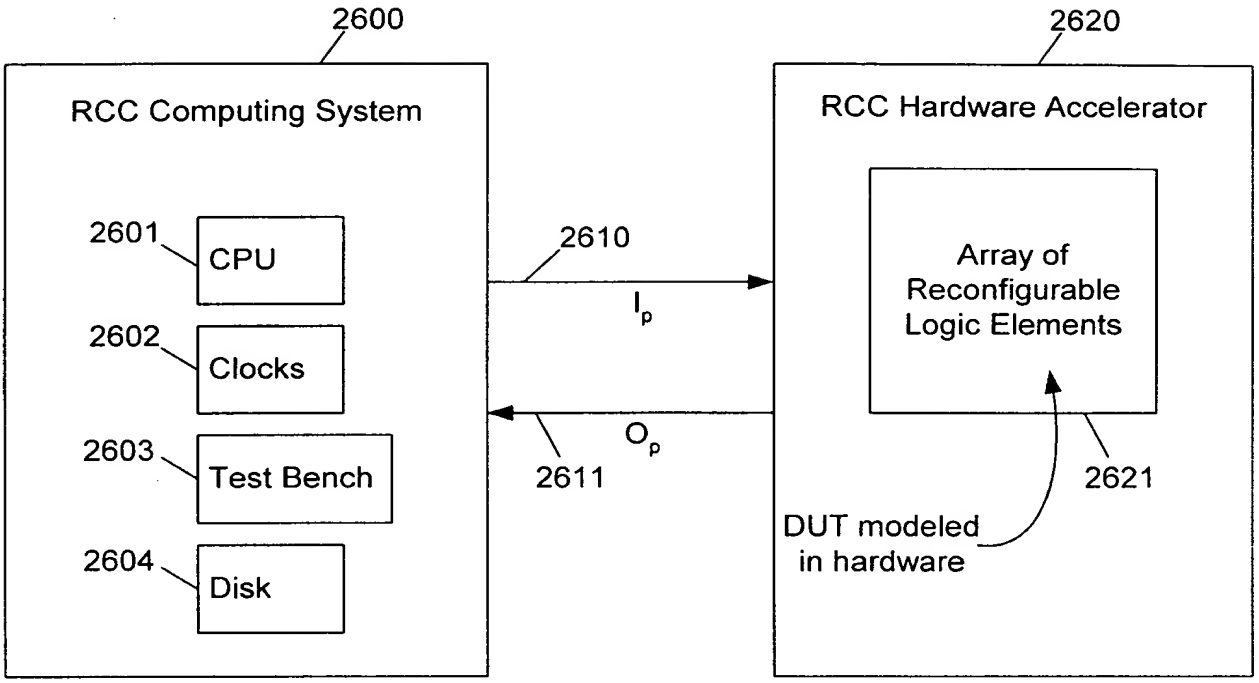


FIG. 83

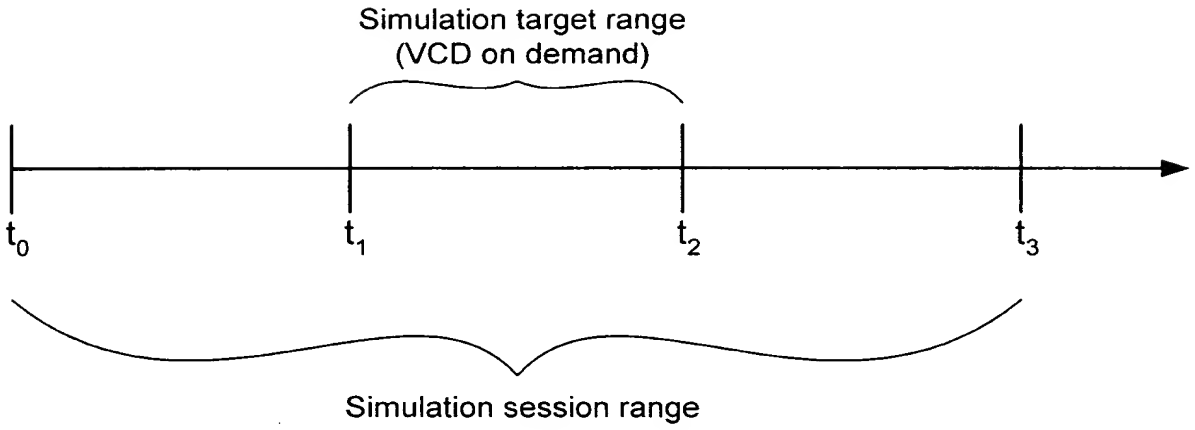

Simulation target range
(VCD on demand)
t_0
t_1
t_2
t_3
Simulation session range

FIG. 84